

محاضرات في لغة C

مقدمة عامة على لغة C

* مميزات لغة C :-

- : C
- : Structured Language
- : Bit manipulation ()
- : Bit
- : byte
- : Portable
- : Apple IBM
- : ANSI C

The Basic Structure of c Program

قواعد بناء البرنامج

C

```
# include < stdio .h >
main ( )
{
printf ( " hello egypt " );
}
```

ملاحظات هامة

- # include <h >
- . include
- { main ()
- }
- (=)
- /* */
- { }
- ∴

أنواع البيانات Data Type

:-

*

) integer ()

-

. float (

-

* الجدول التالى يوضح هذه الأنواع وكذلك عدد البايت byte التى يشغلها :-

نوع المتغير	طوله ببايت	الموسم
(char)	1	
(int)	2	32768 - 32768
(long)	4	2014704830648 - 2014704830648
(float)	4	e + 38 e - 38
(double)	8	e + 308 e - 308

وفيما يلى المقصود بكل هذه الأنواع :-

:

-

. ()

:

-

: (Long)

-

5.33

:

-

:

-

:-

* تسميته المتغير :-

-

hp

HP

-

.

-

* الأعلان عن المتغيرات :-

Variables

C

-

C ++

int a ;

float ;

* المؤثرات operators

1- arithmetic operators

addition	+
Subtraction	-
multiplication	*
division	/

2- Relational operators :

النتيجة	مثلي	ليز	لموثر
1	$10 > 8$	$>$	greater than
0	$10 < 8$	$<$	less than
0	$10 == 8$	$==$	equal to
1	$10 != 8$	$!=$	not equal to
0	$10 < = 8$	$<=$	less than or equal to
0	$10 > = 8$	$>=$	greater than or equal to

3- Logical operator

النتيجة	مثلي	ليز	لموثر
1	$10 > 8 \&\& 9 > 7$	$\&\&$	AND
1	$10 < 8 \parallel 7 < 8$	$\parallel$	OR
1	$!(10 == 8)$	$!$	NOT

4- Assignment Operators

9 = 6

النتيجة	لموثر	طريقة لحيثية	النصي لتكالي
11	$+ =$ addition assignment operators	$A + = 5$	$A = a + 5$
1	Subtraiton assignment opertors	$A - = 5$	$A = a - 5$
30	Multiplication assibnment operators	$A * = 5$	$A = a + 5$
2	Division assignment operators	$A / = 3$	$A = a / 3$

5- Decrement & increment

محاضرات في لغة C

	7	A ++	A = a + 1
	5	A --	A = a - 1

6- %

)

1 = 5 / 2

C = a % 2

a = 5

دوال الإدخال والإخراج

* دالة الطباعة على الشاشة printf() *

: _____

#

stdio.h

printf ()

include

include < stdio.h >

printf ()

-

(String , char , float , int)

.

-

1

printf (" welcome with compusciene ")

" "

2

printf (" \n welcome \n with \n compusciene ") ;

100

new line

100

:

welcome

with

compusciene

printf ()

لماذا	لا تستخدم	لماذا
-------	-----------	-------

محاضرات في لغة C

printf (" \n ")	new line	\n
print f (" \t ")	(Tap) 8	\t
print f (" \b ")	Backspace	\b
printf (" x41 ") : a	hexadecimal	\xdd
printf (" \101") : a	octol (each d represents a digit)	\ddd
printf (" \" ")	double quate	\"
printf (" \a ")	()	\a

: *

```

/* program name 1 */
#include <stdio.h>
main ( )
{
printf ( " \n this text display in new line " );
printf ( " \n word1 \t lab1 \t tab2 " );
printf ( " \n bell \a bell \b " );
printf ( " \n this line display quotations \ " " );
printf ( " \n " );
}

```

: *

print f ()

```

printf ( " % d " , a );
printf ( " % f " , b );

```

% -

:

الكود	الاستخدام	مثال
-------	-----------	------

printf (" % d “ , - 10)	(Signed decimal integer) int	%d
printf (“ % p “ , 507)	(floating point) float	%f
printf (“ % c “ , “ a “)	() char Single character	%c
printf (“ % s “ , “ is “)	String	%s
printf (“ % u “ , “ 10 ”)	((unsigned decimal integer)	%u
printf (“ % x “ , af)	hex	%x
printf (“ % o “ , 67)	Qctal	%o

*

```

/* program name 2 */
# include < stdio.h >
main ( )
{
int a , b , c ;
float f ;
long t ;
char ch = ‘y ‘ ;
char name [10] = " aly " ;
a = 5 ;
b = 10 ;
c = a + b ;
printf ( “ \n c = % d “ , c ) ;
printf ( “ \n f = % f “ , f ) ;
printf ( “ \n name = % s “ , name ) ;
printf ( “ \n ch = % c “ , ch ) ;
printf ( “ \n t = % ld “ , t ) ;
}

```

ملاحظة على اولى

محاضرات فى لغة C

stdio.h	# include < stdio.h >	2	•
	printf ()		
{	4 main ()	3	•
		7 , 6 , 5	•
y	(char) ch	8	•
	name	9	•
		Aly	
	C A , B	12 11 10	•
	{		•
534.6735	: % 3F	<u>ملاحظات :</u>	
		534.674	

Print f ()

:

```
# include < stdio.h >
main ( )
{
int a , b , c ;
float r , s , t ;
char name [10];
printf ( " \n \n enter your name : " ) ;
scanf ( " % s " , name ) ;
printf ( " a = " ) ;
scanf ( " % d " , & a ) ;
printf ( " b = " ) ;
scanf ( " % d " , & b ) ;
printf ( " r = " ) ;
scanf ( " % f " , & r ) ;
printf ( " s = " ) ;
scanf ( " % f " & s ) ;
printf ( " \n welcome % s " , name ) ;
printf ( " \n \n c = a + b = % d " , a + b ) ;
printf ( " \n \n t = r + s = % d " , r + s ) ;
}
```

ملاحظات على الحل

a , b , c , r , s , t , name	•
enter your name printf()	•
name scan f ()	•
0000	
scan f () (" % d " , &a) scan f ()	•
a	

ماذا يعنى المؤشر &

a a

: &a

&

- : *

محاضرات في لغة C

```
enter your name : ahmed
a = 5
b = 10
r = 20
s = 30
welcome ahmed
c = a + b = 15
t = r + s = 50
```

تمرين : برنامج لإيجاد مساحة الدائرة

```
# include < stdio .h >
/* program to calculate area of circle */
main ( )
{
float radius, area ;
printf ( “ radius = ? “ ) ;
scanf ( “ % f “ , radius ) ;
area = 3.14159 * radius * radius ;
printf ( “ area = % f “ , area ) ;
}
```

/* function heading */

/* variable declarations */

/* output statement */

/* input statement */

/* assignment statement */

/* output statement */

* دوال إدخال حرف *

:

```
getchar ( ) , getche ( ) , getch ( )
```

(stdio .h) - الدالة getchar ()

enter

مثال

```
char a ;
a = getchar ( ) ;
printf ( “ % c “ , a ) ;
```

(conio .h) - الدالة getche ()

enter

getchar ()

.

مثال

```
char a ;
a = getche ( ) ;
printf ( “ % c “ , a ) ;
```

3 – الدالة getch () : (conio.h)

enter

مثال

```
char a ;
a = getch ( ) ;
printf ( “ % c “ , a ) ;
```

* دالة طباعة حرف واحد putchar () : (stdio.h)

putchar (' a ') -
a

* دالة إدخال عبارة حرفية gets () : stdio.h

gets () -
string

مثال

```
char name[20];
gets (name) ;
```

Name gets ()

* دالة طباعة عبارة حرفية puts () : stdio.h

string -

مثال :

```
char name[10] = “ ahmed “
puts (name) ;
puts ( “mohammed “ ) ;
```

```
ahmed
mohammed
```

ملاحظة : char name[10]

10

* بعض الدوال المكتبة

الدالة	النوع	الغرض	ملف include
abs (i)		i	stdlib.h
log (d)		d	math.h
pow (d1 , d2)		d1 d1	math.h

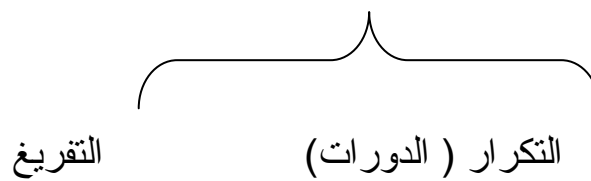
محاضرات في لغة C

math.h	l		labs (e)
math.h	d		sin (d)
math.h	d		sqrt (d)
string.h	s1 s2		strcpy (s1,s2)

أوامر التحكم

main

- :



أولا التكرار Loop

1- الدوارة for

For (initial – value ; condition ; increment) ; statement ;

الصورة العامة :

:

: initial – value

: condition

: increment

9

مثال :

```
# include < stdio.h >
main ( )
{
int i ;
for ( i = 0 ; i < 10 ; i ++ )
printf ( “ \n i = % d ” , i ) ;
}
```

* تنفيذ أكثر من جملة مع For :

```
{
.
}
```

مثال :

```
# include < stdio.h >
/* calculate the average of n numbers */
main ( )
{
```

```
int n , count ;
float x , average , sum = 0 ;
/* initialize and read in a value for n */
printf ( " how many numbers ? " ) ;
scanf ( " % d " , & n ) ;
/* read in the numbers */
for ( count = 1 ; count <= n ; count ++ )
{
printf ( " x = " ) ;
scanf ( " % f " ; & x ) ;
sum + = x ;
}
/* calculate the average and write out the answer */
average = sum / n ;
printf ( " \n the average is % f \n " , average ) ;
}
```

* الدورات المتداخلة باستخدام For

-

loop

-

Loop

12 × 12 1 × 1

مثال :

```
# include < stdio.h >
main ( )
{
int i , j ;
for ( i = 1 ; i <= 12 ; i ++ )
{
printf ( " in " ) ;
for ( j = i ; j <= 12 ; j ++ )
{
printf ( " i * j = % d " , i * j ) ;
}
}
}
```

• الدورة اللانهائية باستخدام For :

For (;;)

CTRL + C

while

2 - الدورة while:

الصورة العامة :

محاضرات في لغة C

While (condition) Statement ;	While (condition) { statement 1 ; statement 2 ; }
------------------------------------	---

()

{ }

•

.

False .

•

False

•

While

تمرين :

```
# include < stdio.h >
main ( )
{
int n , count = 1 ;
float x , average , sum = 0 ;
printf ( " how many number ? " ) ;
scanf ( " % d , & n ) ;
while ( count < = n ) ;
{
printf ( " x = " ) ;
scanf ( " % f " , & x ) ;
sum + = x ;
count ++;
}
average = sum / n ;
printf ( " \n the average is % f " ; average ) ;
}
```

* ملاحظات على For و While *

For -

while

while -

do while : 3- الدوارة while ... Do

Do

.

True ()

False

9 0

:

```
# include < stdio.h >
main ( )
{
```

```
int count = 0 ;
int total = 0 ;
do
{
total += count ;
printf ( " count = % d , total = % d \ n " , count ++ , total ) ;
}
while ( count <= 10 ) ;
}
```

ثانياً : التفريغ :

(1) التفريغ المشروط

* جملة الشرط IF : IF
الصورة العامة

```
( condition )
if ( condition )
statement ;
```

(condition)

ملاحظة

```
{ if
: }
```

```
if ( condition )
{
statement 1 ;
statement 2 ;
}
```

* جملة if الشرطية المتداخلة :

```
if ( condition )
:
if ( condition )
if ( condition )
....
```

* الجملة الشرطية if ...else :

```
if ( condition )
{
statement 1
}
else
{
```

```
statement 2
}
```

```
( statement 1 )      true      ( condition )
if ....else          ( statement 2 )
```

.

* الجملة الشرطية if else :

- :

IF

الطريقة الأولى :

```
i = 5 ;
if ( i < 5 ) ;
    printf ( " i less than 5 " ) ;
if ( i = 5 )
    printf ( " i equal to 5 " ) ;
if ( i > 5 )
    printf ( " i greater than 5 " ) ;
```

الطريقة الثانية

```
if ( condition )
    statement 1 ;
else if ( condition )
    statement 2 ;
else if ( condition )
    statement 3 ;
```

:

```
I = 5 ;
if ( i < 5 )
    printf ( " i less than 5 " ) ;
else if ( i = 5 )
    printf ( " i equal to 5 " ;
else if ( i > 5 )
    printf ( " i greater than 5 " ) ;
```

برنامج : اشرح وظيفة البرنامج مع إيجاد الناتج إذا كانت القيم 13 و 7 على التوالي :-

```
# include < stdio.h >
main ( )
{
float num1 , num2 ;
char op ;
while (1)
```

```
{
printf ( " type number , operator , number \n " );
scanf ( " % f % c % f ) , & num 1 , & op , & num 2 ) ;
if ( op == ' + ' )
    printf ( " = % f " , num1 + num2 ;
else if ( op == ' - ' )
    printf ( " = % f " , num1 - num2 ;
else if ( op == ' * ' )
    printf ( " % f " , num1 * num2 ;
else if ( op == ' / ' )
    printf ( " = % f " , num1 / num2 ;
    printf ( " \n \n " ) ;
}
}
```

2 - التفريع Switch case

Switch

Switch

Switch case

تعيين :

```
#include < stdio.h>
main ( )
{
float num1,num2;
char ch,op;
do
{
printf( "\n type num1 op num2: " );
scanf( "%f %c %f ", &num1,&op,&num2);
switch (op)
{
case "+";
printf ( "sum = %f ", num1+num2);
break;
case "-";
```

```
printf ( "sub = %f ", num1-num2);
break;
case "*";
printf ( "mul = %f ", num1*num2);
break;
case "/";
printf ( "div = %f ", num1/num2);
break;
default:
printf("\n unknowen operator .");
}
printf("\n Again (y/n): ");
}
while (( ch=getch())== 'y' ):
}
```

switch case

ملاحظة :

menu ()

switch

break

(3) التفريغ غير المشروط goto

```
#include <stdio.h>
main( )
{
char ss;
ss:
    printf("\t ALLAH");
    goto ss;
}
```

goto

ملاحظات : -

CTRL+C

-

** دوال تحسين المدخلات والمخرجات **

(1) داله مسح الشاشة clrscr() : ملف التوجيه conio.h

-

clrscr()

-

محاضرات فى لغة C

(2) داله تغيير ووضع المؤشر gotoxy() : ملف التوجيه conio

y x -

gotoxy() -

gotoxy(30,10) 30 :

(3) داله تغيير لون الكتابة textcolor() : ملف التوجيه conio.h

-

-

textcolor(color no)

textcolor(color name)

الجدول التالى لوضع اكواد الالوان وأسمائها

اللون	رقم اللون	اسم اللون
	0	BLACK
	1	BLUE
	2	GRGEN
	3	CYAN
	4	RED
	5	MAGENTA
	6	BROWN
	7	LIGHTGRAY
	8	DARKGRAY
	9	LIGHBLUE
	10	LIGHTGREEN
	11	LIGHTRED

محاضرات فى لغة C

LIGHTRED	12	
LIGHTMAGENTA	13	
YELLOW	14	
WHITG	15	

مسائل :

1- اكتب برنامج لإيجاد جملة مبلغ ما اودع فى بنك لمدة 20 سنة بفائدة 6% لمبلغ \$ 100 والمعادلة كما يلى :

$$f=p(1+i)^n$$

حيث: n عدد السنوات p المبلغ I معدل الفائدة
** البرنامج **

```
#include <stdio.h>
#include <math.h>
main()
{
float p,r,n,i,f;
/* read input data */
printf( "please enter a value for the princepal (p) : " );
scanf("%f", &p);
printf("please enter a value for the interest rate( r) : " );
scanf("%f", &r);
printf("please enter a value for the number or year ( n) : " );
scanf("%f", &n);
/*calculate i , then f */
i=r/100
```

```
f=p*pow((1+i),n);
/* write output */
printf( "\n the final value (f) is : %2f \n",f);
}
```

2- اكتب برنامج لإيجاد الجذور الحقيقية لمعادلة من الدرجة الثانية
 $ax^2 + bx + c = 0$
 بمعلومية
 وباستخدام الصيغة

$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

** **

```
# include <stdio.h>
# include < math.h>
/* real roots of a quadratic equation */
main( )
{
float a,b,c,d,x1,x2;
/* read input data */
printf("a= ");
scanf("%f",&a);
printf("b= ");
scanf("%f",&b);
printf("c= ");
scanf("%f",&c);
```

```
/* carry out the calculations */
d=sqrt(b*b - 4*a*c);
x1=(-b+d)/(2*a);
x2=(-b-d)/(2*a);
/* write output */
printf("x1= ",%e    x2=%e ",x1,x2);
}
```

ملاحظة :

: %e

3- اكتب برنامج لتقويم كثيرة الحدود

$$y = \left(\frac{x-1}{x}\right) + \frac{1}{2}\left(\frac{x-1}{x}\right)^2 + \frac{1}{3}\left(\frac{x-1}{x}\right)^3 + \frac{1}{4}\left(\frac{x-1}{x}\right)^4 + \frac{1}{5}\left(\frac{x-1}{x}\right)^5$$

$$u = \left(\frac{x-1}{x}\right)$$

∴ *

$$y = + \frac{u^2}{2} + \frac{u^3}{3} + \frac{u^4}{4} + \frac{u^5}{5}$$

** **

```
# include<stdio.h>
# include<math.h>
main( )
{
float u,x,y;
/* read input data */
printf(" x = ");
scanf("%f",&x);
/* carry out the calculations */
```

```
u= (x -1)/x;
y=u+pow(u,2.)/2+pow(u,3.)/3+pow(u,4.)/4+pow(u,5.)/5;
printf("x= ",%f      y ="%f ", x,y );
}
```

4- برنامج لإدخال كلمة سر

```
# include<stdio.h>
# include<conio.h>
main( )
{
char pass[10];
do
{
printf("\n enter password: ");
scanf("%s",pass);
}
while(strcmp(pass,"dahe")!=0);
}
```

:

string

: strcmp()

•

•

تعديل لبرنامج كلمة السر:-
(عدم ظهور كلمة السر التى يكتبها المستخدم على الشاشة)

```
# include<stdio.h>
# include<conio.h>
main( )
{
  chat ch;
  char pass[10];
  do
  {
    textcolor(WHITE);
    textbackground(BLUE);
    cprintf("\n enter password: “ );
    textbackgroundf(WHITE);
    cscanf(“%s”,pass);
  }
  while(strcmp(pass,”dahe”)!=0);
}
```

اكتب برنامج:
لطباعة عبارة حرفية تم ادخالها مع بيان عدد حروفها وعدد الكلمات

```
/* count characters and word in a phrase      typed in */
# include <stdio.h>
main( )
{
  int charcnt=0;
  int wordcnt=0;
  char ch;
  printf(“ type in a phrase : \n “);
  /* read characters and quit loop on [ return ] */
  while((ch=getche())!='\r');
  {
    charcnt ++          /* count character */
    if(ch==' ');        /* space ? */
    wordcnt++;
  }
  printf(“\n character count is %d “,charcnt );
  printf(“\n word count is %d “ , wordcnt );
}
```

تذكر

محاضرات فى لغة C

عثماتؤء	لستعلم لءلة
	printf()
	puts()
	scanf()
	gets()
	getch()
	getche()
	clrscr()
	gotoxy()
	Textcolor()
	textbackground()
	cprintf() , cputs
	cscanf(), cgets

الدوال FUNCTION

الدوال فى لغة الـ C نوعين :-

scanf() printf() Built in Function (1)

- :User Function (2)

-

:

-

.

مزايا استخدام الدوال

-1 :

-2

:

include < filename>

```
function declarations;
main( )
{
    function1
    function2
}
```

```
function1
{
-----
-----
}
```

```
function2
{
-----
-----
}
```

أذن تتكون الدالة من :-
(1) الإعلان عن الدالة

(2) جسم الدالة

مثال : دالة بدون دليل

```
#include <stdio.h>
#include<conio.h>
void line2(void);
main( )
{
clrscr( )
line2( )
printf(“ ** Allah the god of all world ** \n “);
line2( )
/* end of main( ) function */
}
```

```
void line2(void)
{
int j;
for(j=0;j<=40;j++);
printf( “ * “);
printf(“\n “);
}
```

- :

line2()

line2()

الموضع الأول : function declaration

main() 3

.

الموضع الثانى : main()

function coling line2() 9 7

.

الموضع الثالث : .main()

function definition

{ 11

}

**** أنواع الدوال Function Type ****

int function	
float function	
string function	
void function	
struct function	structure

تمرين :

```
# include <stdio.h>
int sum(int a, int b )
main( )
{
int z , x = 10 , y = 40;
z = sum(x,y);
printf(“\n\n z = %d “ , z );
}
/*      */
int sum(int a , int b )
{
int s;
```

```
s = a + b ;
return s;
}
```

**** ملاحظات على البرنامج ****

int	sum()	2	-
.		void	
)	x ,y	6	-
	(		
	-:	14 9	-
.10	x	a	9
	.40 y	b	
s	b	a	12
return	s		13
.	z = s	6	z = sum(x,y)
			:

	int	int
	float	float
void	(return	)

**** استدعاء الدالة ****

void line2(void)

*

*

```
# include < stdio.h>
# include <conio.h>
void line3(int no)
main( )
{
clrscr( )
line3(30);
printf(“ ** Allah the god of all world ** \n “);
line3(70);
}
```

```
/*      */
```

```
void line3(int no)
```

```
{
```

```
int j , no;
```

```
for(j = 0 ; j <= no ; j++ )
```

```
printf(" * ");
```

```
printf("\n");
```

```
}
```

```
.no
```

```
:
```

```
**
```

```
.
```

**** استدعاء الدالة بمتغيرات ****

```
-
```

```
-
```

```
( )
```

*** برنامج لتحديد الكمية الأكبر من ثلاث كميات صحيحة**

```
# include <stdio.h>
```

```
/* determine the largest of three integer quantities */
```

```
main( )
```

```
{
```

```
int a, b, c, d;
```

```
/* read the integer quantities */
```

```
printf("\n a = ");
```

```
scanf( % d " , &a );
```

```
printf("\n b = ");
```

```
scanf( % d " , &b );
```

```
printf("\n c = ");
```

```
scanf( % d " , &c );
```

```
/* calculate and display the maximum value */
```

```
d = maximum( a, b );
```

```
printf("\n \n maximum = % d , maximum(c ,d ));
```

```
}
```

```
/* determine the larger of two integer quantities */
```

```
maximum(x ,y )
```

```
int x,y;
{
int z;
z = (x >= y) ? x : y;
return(z);
}
```

ملاحظة هامة : (? |) - :

```
test (x >= y) -
if -
Else | -
x (x >= y)
y
```

**** باستخدام الدوال أكتب برنامج لتحويل الساعات إلى دقائق مع إيجاد الفرق**

```
#include<stdio.h>
/* calculates difference between two times */
main( )
{
int mins1 , mins2;
printf(" type first time ( from 3:22 ): ");
mins1=getmins();
printf(" type second(later) time ");
mins2=getmins();
printf(" Difference is %d minutes." , mins2-mins1);
}
/* getmins function */
/* gets time in hours : minutes formal */
/* return time in minutes */
getmins( )
{
int hours, minuts;
scanf(" %d : %d ", &hours , &minutes);
return(hours * 60 + minutes);
}
```

*** الآتى برنامج يستعمل دالة من نوع int تتكون من دالتين :-**

-

```
#include <stdio.h>
int sqr(int a):
int qup(int q):
main( )
{
int s , qu , no =10;
s = aqr(no);
qu = qup(no);
printf("\n squer of no = %d " , s ):
printf("\n qupic of no = %d " , qu ):
}
int sqr(int a)
{
int v1;
v1= a*a;
return v1;
}
int qup(int q )
{
int v2;
v2=q*q*q;
return v2;
}
```

:

```
squar of on = 100
qupic of on = 1000
```

الآتي برنامج يستعمل دالة من نوع float تقوم بجمع رقمين وإعادة النتيجة الى الدالة الرئيسية عند الاستدعاء . والقيمة المرتجعة قيمة حقيقية float وبالتالي يكون نوع الدالة float

```
#include <stdio.h>
#include <conio.h>
```

```
float add( float x , float y )
```

```
main( )
{
float no1, no2;
printf("\n enter no1 , no2 : );
scanf(" %f ,%f " , &no` ,&no2 ):
```

```
printf("\n addition of number is %f", add(no1,no2) );  
}
```

```
float add(float x , float y );  
{  
float yt;  
yt= x + y ;  
return yt;  
}
```

-:

Enter no1 ,no2 : 3,2 , 4,3
Addition of squares is 6.5

**** الآتي برنامج لإيجاد العدد فردى أم زوجى وكذلك موضحا العدد سالب أم موجب**

```
# include < stdio.h>  
/* tests use of external variables */  
  
int keynumb;  
  
main( )  
/* external variables */  
{  
printf(" type keynumb : " );  
scanf("%d " , &keynumb );  
addeven( );  
negative( );  
}  
  
/* checks if keynumb is odd or even */  
oddeven( )  
{  
if (keynumb % 2 )  
printf(" keynumb is odd. \n");  
else  
printf( " keynumb is even. \n");  
}
```

```
/* checks if keynumb is nagative */
ngative( )
{
if (keynumb < 0 )
printf(“ keynumb is negative . \n”);
else
printf( “ keynumb is positive . \n”);
}
```

الماكرو MACROS

```
function
(
)
C
:
```

كيفية إنشاء الماكرو :-

```
#define -
preprocessor directive -
```

* الصورة العامة # define macro line

```
#define a 5
5 a
```

تمرين يوضح كيفية الإعلان عن الماكرو وكيفية استعماله

```
# include<stdio.h>
# define sum(a,b) a+b
# define mul(x,y) x*y
```

```
main ( )
{
```

محاضرات في لغة C

```
int v1=5 , v2 = 10;
printf("\n\n sum(v1,v2) = % d",sum(v1,v2);
printf("\n\n mul(v1,v2) = % d",mul(v1,v2);
}
```

ملاحظات على البرنامج :-

```
sum define 2 -
a+b a,b
x*y x,y 3 -
-
```

ملاحظ هامة : a,b

.....)

(

* الفرق بين الماكرو وبين الدالة :-

المرحلة الأولى : source code

المرحلة الثانية : compilation

.obj

المرحلة الثالثة : object

linking .exe

:-

-1 .

-2 object compilation

linking

-3

7)

sum(v1,v2) 15

(15

:. مزايا الماكرو :-

-1

-2

-3

:

```
#include<stdio.h>
#define pi 3.14159
main ( )
{
float area(float);
float radius;
printf("enter radius sphers : ");
scanf("%f", &radius);
printf("area of sphere is 2f",area(radius));
}
/* returns ares of sphere */
float area(rad)
float rad;
{
return(4*pi*rad*rad);
}
```

المصفوفات ARRAYS

*** مقدمة ***

float int char

(int , float , char)

(Type dived data)

.arrays

**** المصفوفة ****

array

int

. float

*** الإعلان عن المصفوفة ***

(1) :

(2)

(3)

120 :

float .

float salary[120] :

.

:- **

A[2 5 9 12 15] :

A

**** الإشارة إلى عناصر المصفوفة ****

[] -

-

1

student 3 75 : -

student[3] = 75 :

salary 38 -

a = salary[37] :

-

printf(" The highest mark is %d ", student[45]);

:- **

include <stdio.h>

define max 10

void main(void)

{

float salary[max];

float average;

int count;

average = 0.0;

for(count=0; count<max ; count++)

```
{
printf(" please enter salary for employee %d", count+1);
scanf("%f", &salary[count]);
}
printf("\n\n salary average is : %f", average/max);
}
```

ملاحظات على البرنامج :-

```
10
.
float salary[max]
float salary[10]
average
scanf() 10 for
scanf("%f ", &salary[count]
average
average += salary[count]
```

الملاحظة الهامة :
for count
scanf("%f", salary[count]) .

count

**** تهئية المصفوفة عند الاعلان عنها ****

```
int marks[5] = { 12 4 7 9 11}
```

int

*** المصفوفة الغير محددة العدد ***

```
int marks[ ] = { 5 9 4 10 7 } :
```

{ }

**** برنامج ****

```

#include <stdio.h>
/* calculate the average of a numbers , the compute deviation of each number
about the average */
main( )
{
int n , count ;
float avg, d , sum=0 ;
/* read in a value for n */
printf("\n how many numbers will be average? ");
scanf("%d", &n);
printf("\n");
/* read in the numbers and calculate ther sum */
for(count = 0 ; count < n count++ )
{
printf("i = %d   x= ", count+1);
scanf("%f", &list[count]);
sum+= list[count];
}
/* calculate and write out the average */
avg = sum/n;
printf("\n the average is %5.2f\n\n ", avg);
/* calculate and write out the deviations about the average */
for(count = 0 ; count < n ; count++)
{
d=list[count] - avg ;
printf("i = %d   x%5.2f   d=%5.2f \n" , count+1, list[count] , d);
}
}

```

ناتج هذا البرنامج :-

The average is 4.18

i = 1	x = 03.00	d = -1.18
i = 2	x = -2.00	d = -6.18
i = 3	x = 12.00	d = 7.82
i = 4	x = 4.40	d = 0.22
i = 5	x = 3.50	d = -0.68

**** برنامج إعادة ترتيب قائمة من الأعداد ****

```

#include<stdio.h>
#define size 100
/* reorder a one-dimensinal, integer array from smallest to largest */
main( )
{
int i , n , x[size];
void reorder(int n , int x[ ]);
/* read in the a value for n */
printf("\n how many number will be entered ? ");
scanf("%d", &n );
printf("\n");
/* read in the list of number */
for(i = 0 ; i<n ; i++)
{
printf("i = %d   x = ", i+1);
scanf("%d", &x[ i ]);
}
/* reprder all array elements */
reorder(n , x )
/* display the reordered list of numbers */
print("\n\reordered list of number :\n\n");
for(i=0;i<n; i++)
printf("i=%d   x=%d\n", i+1 , x[ i ] );
}
int i, item ,temp;
for(item = 0;item<n-1 ; item++ )
if(x[ i ] < x[item])
{
/* interchange twwo elements */
temp = x[item];
x[item] = x[ i ];
x[ i ] = temp ;
}
return;
}

```

int a[5] [10]

10 5

for

ملاحظة :

* إعطاء قيمة ابتدائية للمصفوفة:

int a[3][4] =

```
{
  {4,5,7,8 },
  {3,2,4,5},
  {7,8,9,6}
};
```

5 0,1 4 0,0

ملاحظة :

{ }

000

{ }

شكل آخر للمصفوفة ذات البعدين :

int values[3][4] = {1,2,3,4,5,6,7,8,9,10,11,12};

2 0,1 values[0][0] 1 0,0

values[0][1]

مثال: على استخدام المصفوفات متعددة الأعمدة

include <stdio.h>

define stud 3

define marks 10

void main (void)

{

int student[stud][marks];

int c1 ,c1 ;

for(c1 = 0 ; c1 < stud ; c1++)

for (c2 = 0 ; c2 < marks ; c2++)

{

printf (“ enter marks: %2d for student %2d : “ , c2+1 , c1+1);

scanf (“ %d “ , &student[c1][c2]);

}

for(c1 = 0 ; c1 < stud ; c1++)

for(c2= 0 ; c2 < marks ; c2++)

printf(“ student [%2d] , marks[%2d] = %d\n “, c1+1, c2+1 , student[c1][c2]);

}

```
int student[stud][marks];
```

```
for
```

* برنامج : جمع جدولين من العداد اى نحسب مجموع العناصر المتناظرة
فى الجدولين اى نحسب

```
c[i][j] = a[i][j] + b[i][j]
```

ثم نستخرج الجدول الجديد محتويًا على المجموع

```
# include <stdio.h>
# define mrow 20
# define mcol 30
/* caluculates the sum of the elements in two tables of integers */
void main (void)
{
int mrow , mcol;
/* array definitions */
int a[mrow][mcol] , b[mcol][mcol] , c[mcol][mcol];
/* function protyotypes */
void readinput(int a[ ][mcol],int nrow , int ncol);
void computesum ( int a[ ][mcol], int b[ ][mcol] , int c[ ][mcol], int nrow ,
int ncol);
void writeoutput ( int c[ ][mcol], int nrow , int ncol );

printf ( " How many rows ?" );
scanf ( " %d " , nrow);
printf ( " How many columns ?" );
scanf ( " %d " , ncol);

printf ( " \n\n first table : \n" );
readinput( a, nrow , ncol );

printf ( " \n\n second table : \n" );
readinput( b, nrow , ncol );

computsum(a, b, c, nrow, ncol);
printf ( " \n\n sums of the elements : \n \n" );
writeoutput(c , nrow , ncol);
}
```

```
void readinput( int a[ ][mcol] , int m , int n )
```

```
/* read in a table of integers */
{
int row , col ;
for(row = 0 ; row < m ; row ++ )
{
printf("\n enter data for row no. %2d \n " , row+1 );
for( col = 0 ; col < n ; col++)
scanf(" %d " , &a[row][col]);
}
return;
}
```

```
void computsum(int a[ ][mcol] , int b[ ][mcol] , int c[ ][mcol] , int m ,int n )
```

```
/* add the elements of two integer tables */
{
int row, col ;
for(row = 0 ; row < m ; row++)
for(col = 0 ; col < n ; col++)
c[row][col] = a[row][col] + b[row][col];
return;
}
```

```
void writeoutput(int a[ ][mcol] , int m , int n )
```

```
/* write out a table of integers */
{
int roe , col ;
for (row = 0 ; row <m ; row ++ )
{
for(col = 0 ; col < n ; col ++ )
printf("%4d " , a[row][col]);
printf("\n");
}
return;
}
```

ملاحظات :

a ,b ,c -

row -

ncol -

row -

col -

* أفرض ان البرنامج استخدم فى جمع الجدولين الاتى :-

10	11	12	13
14	15	16	17
18	19	20	21

1	2	3	4
5	6	7	8
9	10	11	12

والاتى تنفيذ البرنامج :-

how many rows? 3

how many columns? 4

first table :

enter data for row no. 1

1 2 3 4

enter data for row no. 2

5 6 7 8

enter data for row no. 3

9 10 11 12

second tables :

enter data for row no. 1

10 11 12 13

enter data for row no. 2

14 15 16 17

enter data for row no. 3

18 19 20 21

sums of the elements :

11 13 15 17

19 21 23 25

16 19 31 33

* سلاسل الحروف * (مصفوفة العبارة الحرفية array of string)
string

* الإعلان عن سلاسل الحروف *

```
char name[80] : char **
               c ) 79 name -
                               (null)
char name[ ] = "ali ahmed" **
```

** استخدام مصفوفة الحروف :-

```
printf
gets
```

** والبرنامج التالي يوضح استخدام مصفوفات الحروف :-

```
# include<stdio.h>
void main(void)
{
char namr[80]
printf("place enter your name : ");
gets(name);
printf("welcome , %s ", name);
}
```

*** ملاحظات :**

```
gets name -
name[0] = 'a' -
ss = name[15];
```

**** مصفوفة الحروف تنتهي بالرقم صفر ****

(null)

* *

```
#include <stdio.h>
void main(void)
{
char strin[ ] = "hello";
printf("character array = %s , its length = %d \n, strin , sizeof(string));
}
```

* ملاحظات :

string -

char string[] = "hello" hello -

printf -

sizeof -

character array = hello its length = 6

hello 5 6 -

:

1	2	3	4	5	6
h	e	l	l	o	\0

-: **

```
#include<stdio.h>
void main(void)
{
int count;
char string[ ] = " hello "
for(count = 0 ; count < sizeof(string); count ++)
printf(" string[%d] = %d %c \n ", count, string[count], string[count]);
}
```

for string -

Count < sizeof(string) . -

string -

-:

String[0] = 104	h	String[1] = 101	e
String[2] = 108	l	String[3] = 108	l
String[4] = 111	o	String[5] = 0	

=

**** بعض الاجراءات التى تتعامل مع مصفوفات الحروف ****

(1) معرف عدد الحروف فى المصفوفة :-

strlen

len = strlen("count this string ") -

(2) نسخ مصفوفة حروف الى اخرى :-

strcpy (string1 , string2) : -

string1 string2 -

strcpy -

strcpy(string , " hello my string "); -

string -

(3) دمج محتويات مصفوفتى حروف :-

strcat(string1 , string2) : -

string1 string2 srcat -

string2 string1 -

```
#include <stdio.h>
void main(void)
{
char string1[20] = "learning c ";
char string2[ ] = " is easy ";
strcat(string1 , string2 );
printf(string1);
}
```

learning c is easy :

(4) مقارنة محتويات مصفوفتين :

```

: strcmp
result = strcmp(string1 , string2 );
result strcmp
( )
result
result

```

* البرنامج التالى يعطى مثلاً على استخدام الاجراء strcmp :-

```

#include <stdio.h>
#include <string.h>
void main(void)
{
char string1[80];
char string2[80];
int result;
printf(" enter the first string ");
gets(string1);
printf(" enter the second string : ");
gets(string2);
result = strcmp(string1,string2);
if(result == 0 )
printf("the two string are identical : );
else
printf("the two string are different : );
}

```

ملاحظات :

strcmp -

strcmp -

result = strcmp(string1,string2);

* برنامج عام *

للاستخدام الاجراءات المختلفة التى تتعامل معها مصفوفة الحروف

```

#include <stdio.h>
#include <string.h>
void main(void)
{
char string1[80];
char string2[80];
int result , choice;

```

```
printf(" enter the first string : ");
gets(strig1);
printf("enter the second string : ");
gets(strung2);
printf("\n enter the function you want: \n "
    "1- strcpy \n "
    "2- strcat \n "
    "3- strcmp \n");
printf("enter you choice : );
scanf("%d " , &choice);
switch(choice)
{
case 1:
    printf("\n copy second string to first \n");
    strcpy(string1 , string2 );
    printf(string1);
break;
case 2:
    printf("\n merging the two string \n");
    strcat(string1 , string2 );
    printf(string1);
break;
case 3:
    printf("\n comparing the two strings \n ");
    result = strcmp(string1,string2);
    if(result == 0 )
        printf( " the two string are identical ");
    else
        printf( " the two string are different ");
break;
default :
printf("plase enter a number 1-3 ");
}
}
```

**** تمرير مصفوفة الى الاجراءات ****

-

-

void fn(int num[]);

int

fn

fn(num);

*** البرنامج التالي :-**

يعطى مثلاً على كيفية تمييز مصفوفة حروف الى احد الاجراءات :-

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
void changetoupper(char string[ ]);

void main(void)
{
    char string[80];
    do
    {
        printf(" enter a string to change to upper " case \n enter \ "quit \ "to exit");
        gets(string);
        changetoupper(string);
    }
    while(strcmp(string , "quit"));
}

void changetoupper(char string[ ] )
{
    unsigned int count;
    for(count = 0 ; count < strlen(string); count++)
        printf("%c" , toupper(string[count]));
    printf("\n\n");
}
```

ملاحظات :

—

do...while

—

quit

changetoupper

—

1

toupper

—

```
printf("%c", toupper(string[count]));    printf
```

printf

string functions بعض دوال العبارات الحرفية

وظائفها	لدلة
() ()	strcat()
	strchr()

محاضرات فى لغة C

()	strcmp()
	strcpy()
	strlen()
	strupr()

**** بعض دوال للتحويل من متغير رقمى الى حرفى والعكس :-**

وظيفة	لدلة
	atof
	Atoi
	Atol
	Strtod
	_itoa
	_ltoa

التركيب (المنشآت) Structure

structure

معناه :

.

.

.()

-

:

(char)

•

(int)

•

(float)
structure

* الاعلان عن التركيب :-

```
struct
{
    -----;
    -----;
    -----;
};
```

C struct

```
struct employee
{
    char name[40];
    char address[40];
    int age;
    float salary;
};
```

() employee

```
struct employee emp1;
( )
```

emp1

86 byte emp1
(4 byte) * (2 byte) * (40 byte) * (40 byte) *

* استخدام التركيب :-

```
emp1.age :
age 30
emp1.age=30;
a.emp1.salary; :
```

gets

gets(emp1.name); :

emp1

name

* البرنامج التالى يوضح كيفية الإعلان عن التركيبات واستخدامها :-

include <stdio.h>

void main(void)

{

struct employee

{

char name[40];

char address[40];

int age;

float salary;

}

struct employee emp1;

printf(" enter name :");

gets(emp1.name);

printf("enter address: ");

gets(emp1.address);

printf(" enter age : ");

scanf("%d", &emp1.age);

printf("enter salary : ");

scanf("%d",&emp1.salary);

printf("\n\n you entered: \n ");

printf(" employee name :\\t%s\\n ", emp1.name);

printf(" employee address :\\t%s\\n ", emp1.address);

printf(" employee age :\\t%d\\n ", emp1.age);

printf(" employee salary :\\t%f\\n ", emp1.salary);

}

* مصغوفات من التركيبات *

100

struct employee

{

char namr[40];

```

char address[40];
int age;
float salary;
};
struct employee emp[100]
                emp                employee                -
                                100
                .                -
emp[4].age=35;                -
                                emp[count].salary = 123.5                -
                                count

```

* البرنامج الآتي يعطى مثالاً على كيفية إنشاء مصفوفة من التركيبات والتعامل مع العناصر المختلفة فيها :-

```

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define max 10          //
void main(void)
{
    struct booktag
    {
        char title[40];
        char author[40];
        int pages;
        int year;
        char edition;
    };
    struct bootag book[max];    //
    int c ,c1 , result;
    char temp[10];
    //
    for(c = 0 ; c < max ; c++)
    {
        printf(" enter data for record no[%d]\n "m c+1);
        printf(" or enter \ " quit \ " to end \n\n");
        printf"enter title : " );
        gets(boo[c].title);
        /*                (      )
        result = strcmp (book[c].title,"quit");

```

```

if (result == 0 )
break;
printf("enter author : ");
gets(book[c].author);
printf("enter number of pages : " );
gets(temp);
book[c].pages = atoi(temp);
printf("enter year of publication : " );
gets(temp);
book[c].year = atoi(temp);
printf("enter edition : " );
gets(temp);
book[c].edition = atoi(temp);
}
//
printf("record no. %d\n", c1++);
printf("book title : \t%\n", book[c1].title);
printf(" book author : \t%d\n", book[c1].author);
printf("no of pages :\t%d \n", book[c1].pages);
printf("date of pub. \t%d\n",book[c1].year);
printf("edition : \t%d\n\n",book[c1].edition);
}

```

**** ملاحظات على البرنامج ****

book -

-

booktag -

- :

* * * * *

struct booktag boo[max] 10 -

for -

quit 10 -

quit -

break

```

result = strcmp(book[c].title , "quit");
if(result == 0
break;

```

quit

strcmp

-

fot

result

strcmp

****تمرير التركيب إلى الإجراءات****

-:

.

```

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct studenttag
{
    char name[40];
    int age;
    int grade;
}
void print(struct studenttag student);
void main(void)
{
    struct studenttag student;
    char temp[10];
    printf("enter student name : ");
    gets(student.name)
    printf("enter student age : ");
    gets(temp);
    student.age=atoi(temp);
    printf("enter student grade : ");
    gets(temp);
    student.grade = atoi(temp);
    printf("\n\n you entered: \n\n");
    //
    printf(student);
}
//
void print(struct studenttag student)
{
    printf("student name : \t%s\n ", student.name);
    printf("student age : \t%d \n ", student.age);
    printf("student grade : \t %d\n ", student.grade);
}

```

*** ملاحظات ***

-

print

studenttag

-

```

* * * -:
print
void print(struct studenttag student)
    print(student);
( )
**
.
.

```

**** برنامج ****

```

//
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

struct studenttag
{
    char name[40];
    int age;
    int grade;
};

void print(struct studenttag student);

struct studenttag detinfo(void)

void main(void)

{
    struct studenttag student;
    student = detinfo( );
    print(student);
}

```

```

//
void print(sturct studenttag student)
{
    printf("\n\n you entedred : \n\n ");
}

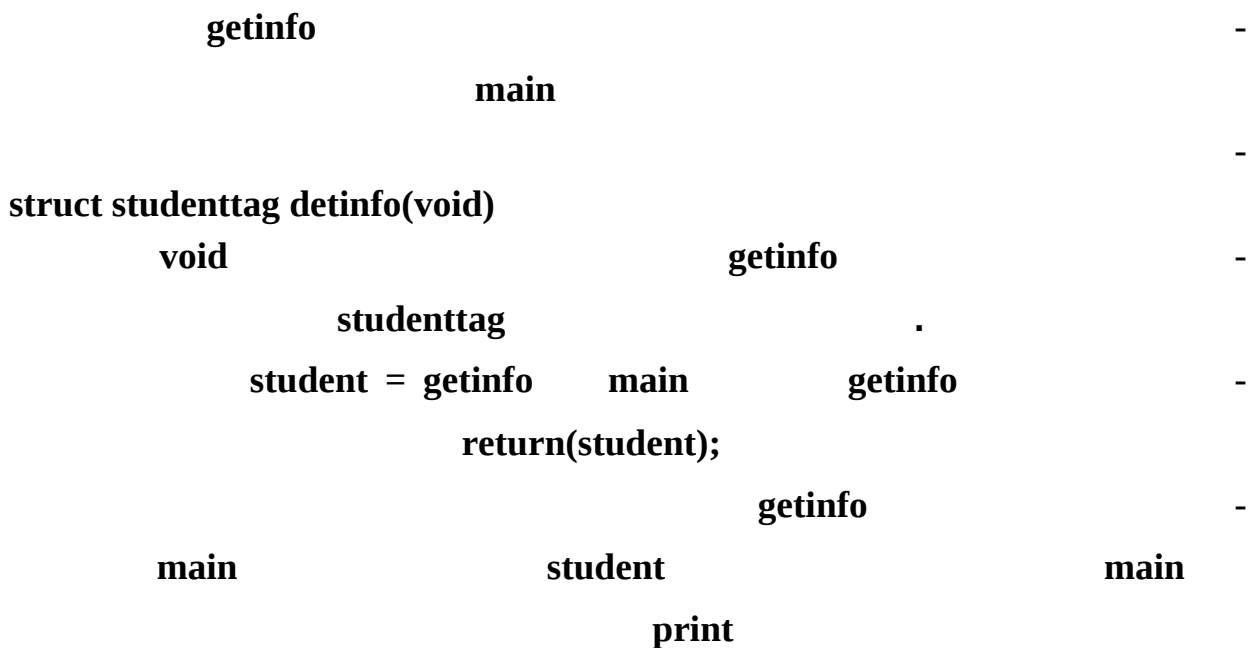
```

```
printf("student name: \t%s\n",student.name);
printf("student age: \t%d\n",student.age);
printf("student grade : \t%d\n",student.grade);
}
```

//

```
struct studenttag getinfo(void)
{
    struct studenttag student;
    char temp[10];
    printf("enter student name : ");
    gets(student.name);
    printf("enter student age : ");
    gets(temp);
    student.age = atoi(temp);
    printf("enter student grade : ");
    gets(temp);
    student.grade = atoi(temp);
    return(student);
}
```

ملاحظات :



**** تركيب داخل تركيب ****

employee

employee

-

```
struct employee
{
char name[40];
char address[40];
int age;
float salary;
}
```

-:

-

```
struct dept
{
int deno;
int product;
sturct employee emp;
}
```

dept

emp

-

struct dept dept1;

dept

-

dept1.emp.age = 25 :

-

dept1

emp

age

**** برنامج ****

```
#include<stdio.h>
#include<stdlib.h>
struct employee
{
char name[40];
char address[40];
int age;
float salary;
};
```

struct department

```
{
int depnum;
char product[40];
struct employee emp;
```

};

```

void main(void)
{
struct deparment dept;
char temp[10];
printf("enter dept no : " );
gets(temp);
dept.deptnum = atoi(temp);
printf("enter dep. Product : " );
gets(dept.product);
prints("enter employee name : " );
gets(dept.emp.name);
printf("enter employee address" );
gets(dept.emp.address);
printf("enter employee age : " );
gets(temp);
dept.emp.age = atoi(temp);
printf("enter employee salary : " );
gets(temp);
dept.emp.salary = atof(temp)
printf("\n\n you entered \n\n " );
printf("dept no : \t\t%d\n ",dept.deptnum);
printf("dept product : \t\t%s\n ",dept.product);
printf("employee address : \t\t%d\n", dept.emp.address);
printf("employee age : \t\t%d\n",dept.emp.age);
printf("employee salary: \t\t%f",dept.emp.salary);
}

```

ملاحظات :-

	employee	-
employee	department	-
		-
name	gets	
	gets(dept.emp.name);	: emp
printf		-

**** تنويغات فى الاعلان عن التركيبات ****

-: *

- الأولى :

struct employee

```
{
char name[40];
char address[40];
int age;
float salary;
};
```

- الثانية :-

```
struct employee emp;
```

*ملاحظة :-

.

:-

```
struct employee
{
char name[40];
char address[40];
int age;
float salary;
} emp;
```

- emp }

struct employee emp;

-

```
struct employee
{
char name[40];
char address[40];
int age;
float salary;
} emp1 , emp2 , emp3;
```

-

```
struct
{
char name[40];
char address[40];
int age;
float salary;
} emp;
```

:-

-

```
struct
{
char name[40];
char address[40];
```

```
int age;  
float salary;  
} emp = {  
    "ali ahmed"  
    "26 st. cairo "  
    39  
    135.6  
};
```

المؤشرات pointer

*معنى المؤشر pointer :-

· () · -
() -
· -

2 byte ram

int d = 5 ;

5

d

:-

: •

.

: •

****الإعلان عن المؤشر****

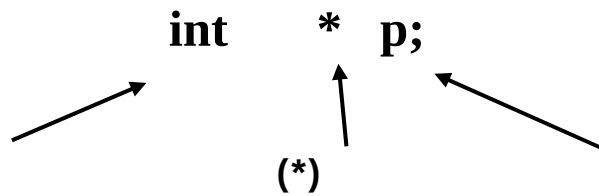
)

(

:-

: الأول

: الثانية



2 byte

p

p

float
degress * degree;

float

: .

```
struct student
{
char name[40];
int mark;
```

}

: .

struct student *st;

****تهيئة المؤشر****

-

&

unary

-

scanf("%d",&var); :

-

int num;

int *ptr;

num = 5;

ptr = #

ptr

num

-

num

5

num

-

num

&

ptr = &num

-

ptr

. () num ptr

-

****استخدام المؤشر** هناك طريقتان لاستخدامه :-**

الطريقة الأولى :

.

-

.

addten(ptr); :

-

num

ptr

addten

-

الطريقة الثانية :

*ptr = 10 ; (*)

-

10 num

-

a=*ptr;

-

تمرين

#include<stdio.h>

void main(void)

{

int u = 3;

int v;

محاضرات في لغة C

```
int *pu;          /* pointer to an integer */
int *pv          /* pointer to an integer */

pu = &u;          /* assing address of u to pu */
v = pu;           /* assign values of u to v */
pv = &v;          /* assing address of v to pv */

printf("\n u=%d  &u=%x  pu=%x  *pu=%d", u , &u , pu , *pu);
printf("\n v=%d  &v=%x  pv=%x  *pv=%d", v , &v , pv , *pv);
}
```

ملاحظات :

v	pv	u	.v	u ,pv	pu	-
		u	3	u		-
			(		) FF8E	
	FF8E (	)	pu	. pu		-
(FF8E			)pu			-
					3	
	v	*pv	3	v		-

**تمرين آخر:

```
#include<stdio.h>
main( )
{
int u1 ,u2;
int v =3;
int pv          / pv pointer to v */
u1 = 2*(v + 5 ); /* ordinary expression */
pv = &v ;
u2 = 2*(*pv + 5 );
printf("\n u1 = %d    us2 = %d ",u1 , u2 );
}
```

ملاحظات :

.	2*(v + 5)	-
v , *pv	2*(*pv + 5)	
	:	-

```
v = 3      u1 = 15
u1=2*(v + 5)
u1=2*(3+5)
u1= 16
```

```
u2 = 16      pv = 3
u2 = 2*(*pv + 5)
u2=2*(3+5)
u2=16
```

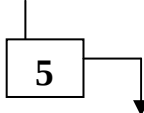
** تمرير المؤشرات الى الاجراءات الفرعية **

```

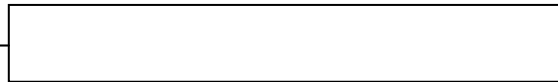
.
C
-
-
-
-: ( " " )

```

```
void main(void)
{
int num = 5;
change(num);
.....;
}
```



```
void change( int num )
{
num = 7 ;
.....;
}
```



ملاحظات :

```
change      5      num      main      -
```

```
5      num      -
```

```
main      num      change      change
```

```
change      main      num      -
```

local

```
) :
```

(

```
-: . *
```

الخطوة الأولى :-

```
void change ( int * ptr );           :
                                     change
                                     :
```

```
void change( int * ptr )
{
    *ptr = 7;
}
```

num	num	ptr	-
-----	-----	-----	---

- :-

before change , num = 5
after change , num = 7

** العودة بأكثر من قيمة من الإجراء الفرعي **

return :-

return

:-

- :-

```
#include<stdio.h>
void fn( int *var1 , int *var2 , int *var3);
void main(void)
{
int var1 ,var2, var3;
fn(&var1 , &var2 , &var3)
printf("var1 = %d , var2 = %d , var3 = %d", var1 , var2 , var3);
}
```

```
void fn( int *p1 , int *p2 , int *p3)
{
*p1=10;
*p2=20;
*p3=30;
}
```

ملاحظات :

var1,var2,var3 -

fn -

p1,p2,p3 fn -

fn -

printf main -

var1,var2,var3

- :- fn -

var1 = 10, var2 = 20 , var3 = 30

** برنامج عام ** (تجديد سجلات العملاء)

-

()

)

(delinquent overdue

.

:

-

.

* توضيح :-

10%

-

-

-

-1

overdue

-2

delinquent

* ويكون البرنامج كما يلي :-

.

(

)

-1

:-

000

-2

1- name

2-street

3- city

4- account number

5- previous balance

6- current balance

7- payment date

:-

-3

-

-

-

-

:

-4

1- name

2- account number

3- street

4- city

5- old balance

6- current balance

7- new balance

8- account status

* فيما يلي محتويات البرنامج :-

```
#include<stdio.h>
```

```
struct date
```

```
{
```

```
int month;
```

```
int day;
```

```
int year;
```

```

};
struct account
{
    char name[50];
    char street[45];
    char city[60];
    int acc_no;
    int acct_type;    /* c = current , o = overdue , d = delinquent */
    float oldbalance;
    float newbalance;
    float paymeny;
    struct date lastpayment;
}

```

```

main( )
{
    int i,n ;

    void readinput(int i );
    void writeoutput(int i );
    printf("customer billing system \n\n");
    printf("how many customer are there ? ");
    scanf("%d",&n);
    for(i=0,i<n,i++)
    {
        readinput(i);
        if(customer[i].payment > 0 )
            customer[i].acct_type =
                (customer[i].oldbalance < 0.1*customer[i].oldbalabce) ? 'o' | 'c';
        else
            customer[i].acct_type =
                (customer[i].oldbalance > 0 ) ? 'd' | 'c' ;
    };
    for (i = 0 ; i < n ; i ++ )
        writeoutput(i);
    }
    void readinput(int i )
    {
        print("\n customer no. %d\n", i+1);
        printf(" name : ");
        sccanf(" %[ ] ",customer[ i ].name);
        printf(" street : ");
        scanf("%[ ] ",customer[ i ].street);
        printf(" city : ");
        scanf(" %[ ] ",customer[ i ].city );
    }
}

```

```

printf(" account number : ");
scanf("%d ", &customer[ i ]. oldbalance);
printf( current payment : );
scanf("%f", &customer[ i ].payment);
printf(" payment date (mm/dd/yyyy) : ");
scanf("%d / %d / %d ", &customer[ i ].lastpayment.month,
                        &customer[ i ].lastpayment.day
                        &customer[ i ].lastpayment.year

return;
}
void writeoutput(int i)
{
printf(" name : %s", customer[i].name);
printf(" account number : %d\n", customer[i].acct_no);
printf("street : %s\n", customer[i].street);
printf("city : %s\n\n", customer[i].city);
printf("old balance " %7.2f", customer[i].oldbalance);
printf(" current payment : %7.2f", customer[i].payment);
printf(" new balance : %7.2f\n", customer[i].newblance);
printf(" account status : ");
switch ( customer[i].acct_type)
{
case 'c'
    printf(" current\n\n");
    break;
case 'o'
    printf(" overdue\n\n");
    break;
case 'd'
    printf(" delinquent\n\n");
    break;
default;
    printf(" error \n\n")
}
return;
}

```