

SCRIPT OF SESSION TFFF chapter

```
#include <stdio.h>
#include <stdlib.h>
unsigned int number=55;
int main() {
    printf("Enter number value:");
    scanf("%d", &number); IS example flow
    printf("Yes\n"); // number = 55
    printf("No\n"); // number = 66
    return 0; ستقوم المبروك بطبع Yes أو No
}
```

Processor

Syntax: int main() {

```
    if ( Condition ) {
        Print(-);
    }
    return 0;
}
```



int main() {

```
    if ( 55 == number ) {
        Print("Yes\n");
    }
```

```
    if ( 66 == number ) {
        Print("No\n");
    }
```

Input: 99
Output: No

الخطأ
الخطأ
logical error.
due to i) مفهوم
ii) number = 55

return 0;

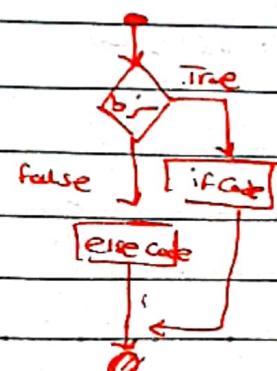
Input: 55
Output: Yes.

if ... else

error detection
صياغة خطأ

Syntax: int main() {

```
    if ( b ) {
        ...
    } else {
        ...
    }
    return 0;
}
```



حالياً
حالياً Processor يجري

جوا على بطء وقف
جوا على بطء وقف
stop
block
block

```
int main() {
    if ( 55 == number ) {
        Print("Yes\n");
    } else {
        Print("No\n");
    }
    return 0;
}
```

Date: _____

Nested if

if (سطح) {

if (سطح) {

مدونة شرح بحد الأقصى للـ loop

فقط ماركة الـ loop (S) .

}

else { }

if elif

Syntax:

if (سطح) {

}

لـ if (سطح) {

elif (سطح) {

}

أي تـ من متـ معـ

else { }

}

ex: alphabet cyclic

Complete condition if ((55-number) && (number >= 6))

{ printf("%d\n"); }

}

Notes

①

else { }

}

more rules حالـ حـاـضـ

automotive

Date: _____

Page: _____

Logical Operators

AND

&&

OR

||

Not

!

AND &&:

returns TRUE when all conditions are true. true بمعنى كل دسغت true.

returns FALSE when any one or more than one condition is False. false بمعنى أي دسغت false one condition is False

int a=5;

if ((a==5) && (a!=6) && (a<-56) && (a>1))

{

printf (" welcome\n");

Output: welcome

}

OR ||:

returns TRUE when one or more than one condition is true. true بمعنى اي دسغت true.

returns FALSE when all conditions are false. false بمعنى كل دسغت false

int a=5;

if ((a!=5) || (a==9) || (a>-56) || (a>4))

{

printf (" welcome\n");

Output: welcome

}

NOT ! : Returns TRUE when condition is FALSE

and returns FALSE when condition TRUE

$x \leftarrow v$ if v is false
 $x \leftarrow x$ if v is true

int a=5;

if (! (a==6))

{

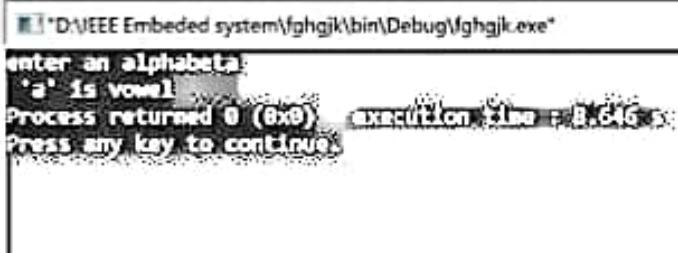
printf (" welcome\n");

Output: welcome

}

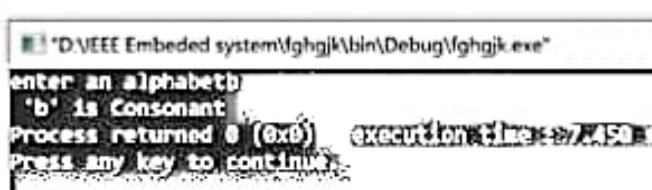
ROX


```
1 //C program to check vowel or consonant
2
3 #include <stdio.h>
4 int main()
5 {
6     char ch;
7     printf("enter an alphabet");
8     scanf("%c",&ch);
9     if((ch=='a') || (ch=='e') || (ch=='o') || (ch=='u') || (ch=='i') || (ch=='A') || (ch=='E') || (ch=='O') || (ch=='U') || (ch=='I'))
10    {printf(" '%c' is vowel",ch);
11    }
12    else if((ch >= 'a' && ch >= 'z') || (ch >= 'A' && ch >= 'Z'))
13    {printf(" '%c' is Consonant",ch);
14    }
15    else
16    {printf(" '%c' is not an alphabet",ch);
17    }
18    return 0;
19 }
20
```



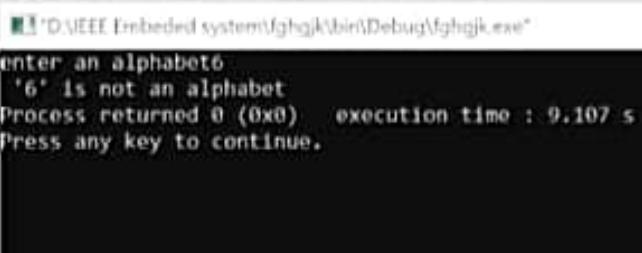
D:\EEE Embedded system\ghgjk\bin\Debug\ghgjk.exe
enter an alphabet:
'a' is vowel
Process returned 0 (0x0) execution time : 0.646s
Press any key to continue:

```
1 //C program to check vowel or consonant
2
3 #include <stdio.h>
4 int main()
5 {
6     char ch;
7     printf("enter an alphabet");
8     scanf("%c",&ch);
9     if((ch=='a')||(ch=='e')||(ch=='o')||(ch=='u')||(ch=='i')||(ch=='A')||(ch=='E')||(ch=='O')||(ch=='U')||(ch=='I'))
10    {printf(" '%c' is vowel",ch);
11    }
12    else if((ch >= 'a' && ch >= 'z')||(ch >= 'A' && ch >= 'Z'))
13    {printf(" '%c' is Consonant",ch);
14    }
15    else
16    {printf(" '%c' is not an alphabet",ch);
17    }
18    return 0;
19 }
20
```



D:\VEEE Embedded system\fghgjk\bin\Debug\fghgjk.exe
enter an alphabet
'b' is Consonant
Process returned 0 (0x0) execution time: 0:00:00.000
Press any key to continue...

```
1 //C program to check vowel or consonant
2
3 #include <stdio.h>
4 int main()
5 {
6     char ch;
7     printf("enter an alphabet");
8     scanf("%c",&ch);
9     if((ch=='a') || (ch=='e') || (ch=='o') || (ch=='u') || (ch=='i') || (ch=='A') || (ch=='E') || (ch=='O') || (ch=='U') || (ch=='I'))
10    {printf(" '%c' is vowel",ch);
11    }
12    else if((ch >= 'a' && ch >= 'z') || (ch >= 'A' && ch >= 'Z'))
13    {printf(" '%c' is Consonant",ch);
14    }
15    else
16    {printf(" '%c' is not an alphabet",ch);
17    }
18    return 0;
19 }
20 }
```



```
D:\IEEE Embedded system\fhgjk\bin\Debug\fhgjk.exe"
enter an alphabet6
'6' is not an alphabet
Process returned 0 (0x0) execution time : 9.107 s
Press any key to continue.
```

Notes:

ما هي الخطأ التي تحدث في الكود التالي؟
الإجابة: خطأ curly braces syntax. خطأ error syntax.

```
if (n==4)  
    { printf ("n is4"); }  
else { printf ("n isnot4"); }
```

int A>5;

~~if ($n == 4$)~~

Printf ("n is %d");

147

```
printf ("%d", n);
```

$$\text{outPut} = \underline{\underline{6}}$$

intn. S;

If ($n = 24$)

PrintF("nis4"); مكتوبوا
n++; إلى if نون الشرط متغير

Printf("id", n);

~~Output = 5~~

٣) خصائص الـ APPLICATIONS (التطبيقات) وهي تختلف عن DATA (بيانات) في أن:

ما يطلب هو أن يكون سلوكه ملائكي ويكون سلوكه متوجه وذلك بحسب الProfessor J.J. Gibson

if (~~else~~)

~~-E printf (- - -);~~

~~else (true) {~~

میتوانی بیت حینه سردا جمال عذہ

Error Syntax

Date: _____

(Switch)

If else if will go Case in if Variable size is small

`switch (variable) {`

Case 1:

/* execute your code */

break;

Case 2:

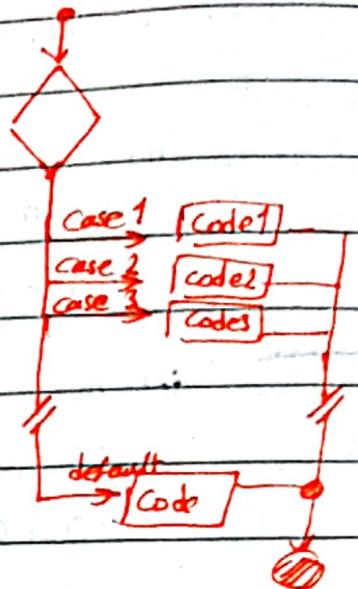
/* execute your code */

break;

default:

/* execute your code */

break;



`switch (useroperator) {`

case '+':

`printf("operator = + \n");`

~~break;~~

case '-':

`printf("operator = - \n");`

That + :

Output: operator = +

~~break;~~

Note:

case '*' :

`printf("operator = * \n");`

~~break;~~

default: `printf("Invalid operator !! \n");`

Operator = +

}

operator = -

operator = *

invalid operator.

Notes: break; less complicated

Input \rightarrow زیر متن منظمه خارج اکسپریس و DefaultSession ⑥

ويمكننا تعيين default للسيمبلات operator على

لذلك في حالة default لا يتحقق SQL Server Cases حيث لا يوجد input لـ IS NULL

~~Siljuska default li break; ja seuraavaksi IS jälkeen~~

Switch (userOperator) {

Case (+):

```
printf ("operator = + (%d)");
```

break;

Case (-1):

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

break

default:

printf ("invalid operator !(\"%s\"");

OutPut: operator= /

case

~~case 'X';~~

input: h

outpt: Invalid Operator !!

Operator = 1

deflation break up is called

printf ("operator != \n");

break;

breaks.

Switch (X) §

Case 1: `printf("X is %1")`,
~~breaks~~

~~Case 1: printf ("X is 1")~~

break

case 2: `printf("x is %d");`

~~breathe~~

Date: _____

Float and Double value is not allowed as a constant value in case label. Only integer constants / constant expressions are allowed in case label. (2)

int main()

```

    float x = 3.14;
    switch (x) {
        case 3.14: printf("x is 3.14");
        break;
        case 1.1: printf("x is 1.1");
        break;
        case 2: printf("x is 2");
        break;
        default: return 0;
    }
  
```

error:
because case label
doesn't reduce to an integer
constant Case 3.14 - X

int main() { int x = 23;

```

    switch (x) {
        case 3+3: printf("choice 1");
        break;
        case 3+1+9: printf("choice 2");
        break;
        default: printf("default");
        break;
    }
  
```

Output: choice 2

Variable expressions are not allowed in case labels. Although macros are allowed (1)

int main() {

int x = 2, y = 2, z = 23;

switch (x) {

case y: printf("number is 2");

break;

case z: printf("number is 23");

break;

default: printf("default case");

break;

ROX

error:

case label doesn't
reduce to an integer
constant.

#include <cs50.h>

#define Y 2

#define Z 23

int main() {

int x = 2;

switch (x) {

case Y: printf("number is 2");

printf("number is 2");

break;

case Z: printf("number is 23");

break;

default:

printf("default case");

break;

Output:**Number is 2**

```

1 //C program to create calculator using switch case and functions
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main()
6 {
7     float a,b;
8     float result=0;
9     char op;
10    printf("enter the operator and the value of two numbers");
11    scanf("%f %c %f",&a,&op,&b);
12    switch(op)
13    {
14        case '+':
15            result= a+b;
16            break;
17        case '-':
18            result=a-b;
19            break;
20        case '*':
21            result=a*b;
22            break;
23        case '/':
24            result=a/b;
25            break;
26        default:
27            printf("invalid operator");
28    }
29    printf("%f = %f %c %f",result,a,op,b);
30    return 0;
}

```

D:\IEEE Embedded system\ghgjk\bin\Debug\ghgjk.exe
 enter the operator and the value of two numbers2+5
 7.000000 - 2.000000 + 5.000000
 Process returned 0 (0x0) execution time : 21.155 s
 Press any key to continue.

Date:

: out - default

[1] ~~#include <stdio.h>~~

```
#include <stdio.h>
int main () {
    int num = 2;
    switch (num + 2) {
        Case 1: printf ("case 1: ");
        Case 2: printf ("case 2: ");
        Case 3: printf ("case 3: ");
        default: printf ("Default: ");
    }
    return 0;
}
```

Output: Default:

[2]

```
#include <stdio.h>
#define L 10
void main () {
    auto a = 10;
    switch (a , a * 2)
        to 20
    {
        Case 1: printf ("ABC"); break;
        Case 1 * 2: printf ("XYZ"); break;
        Case L * 3: printf ("PQR"); break;
        default: printf ("MNO");
        Case L * 4: printf ("WWW"); break;
    }
}
```

Output: XYZ.

Basics of for loop & while loop.

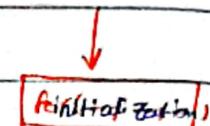
Syntax:

```
for (initialization; condition; increment/decrement)
```

{

Statements;

}



Counter سمعان پر پیش اولین دو کاری

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

```
#include <stdlib.h>
```

```
unsigned short counter=0;
```

```
int main() {
```

```
    for (Counter=1; Counter<=20; Counter++)
```

{

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

}

```
    return;
```

}

Output : Ahmed

Ahmed

20 Ahmed

```
int main() {
```

```
    for ( ; ; )
```

```
{ }
```

```
    return;
```

for. Plain infinite loop. ✗

```
int main() {
```

```
    for (Counter=1; Counter<5)
```

```
{ printf ("Hello world\n");
```

}

Output: Hello world.

ملاحظة

↓

∞

```
int main() {
```

```
    for (Counter=1; Counter<9)
```

```
{ printf ("Hello Ahmed\n"),
```

```
    Counter++;
```

}

Output:

Hello Ahmed

Hello Ahmed

Hello Ahmed

Hello Ahmed

Date: _____

اوليات

```
int main () {
```

```
    for (counter = 100; counter > 0; counter++)
```

```
    {
```

```
        cout << "Ahmed (n);"
```

Ahmed

Ex # Factorial 51

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

```
unsigned int number = 0, factorial = 1;
```

```
int main () {
```

```
    unsigned int loopIterator = 0;
```

```
    cout << "enter the number value \"n\";"
```

```
    scanf (" %i ", &number);
```

```
    if (number <= 0) {
```

factorial is zero or negative (no else)

```
    cout << "error!! factorial of a negative
```

```
    number doesn't exist (n);"
```

```
    else {
```

```
        for (loopIterator = 1; loopIterator <= number;
```

```
            loopIterator++)
```

```
        {
```

```
        factorial *= loopIterator;
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

Enter the number value : 6

```
return 0;
```

Output: Factorial = 720

Date:

Break

Conditions for loop break: loops just (switch & loops) if return
case goes switch loops

① while()

```
{  
    if (break)  
        break;
```

② do {

```
:f(break){
```

```
    break;  
    while ( )
```

③ for(; ;)

```
{  
    if (break)  
        break;
```

```
int main () {  
    for( counter = 1; Counter <= 5; Counter++)  
    {  
        printf (" Ahmed \n ");  
        break;  
    }  
    return 0;  
}
```

Output: Ahmed

int main () {

```
    int n;  
    printf (" enter a number \n ");
```

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

```
    while (n != 0)
```

```
    {  
        if (n < 0)
```

```
            break;
```

```
        printf (" enter a number \n ");
```

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

Output:

enter a number

9

enter a number

-9

loop break

Page: _____

Date: _____

for(; ;)
{ if (b) {

 Continue;
}

{ Continue }

while (b) {

 if (b) {

 Continue;

 }

} {

if (b) {

 Continue;

while (b);

for (Counter = 0; Counter < 5; Counter ++)

{ printf (" Ahmed\n");

 if (Counter == 3) {

 Continue;

 }

 printf (" Mohamed\n");

}

Output:

0 Ahmed

Mohamed

1 Ahmed

Mohamed

2 Ahmed

Mohamed

3 Ahmed

→ Continue;

Ahmed

Mohamed

4 Ahmed

Mohamed.