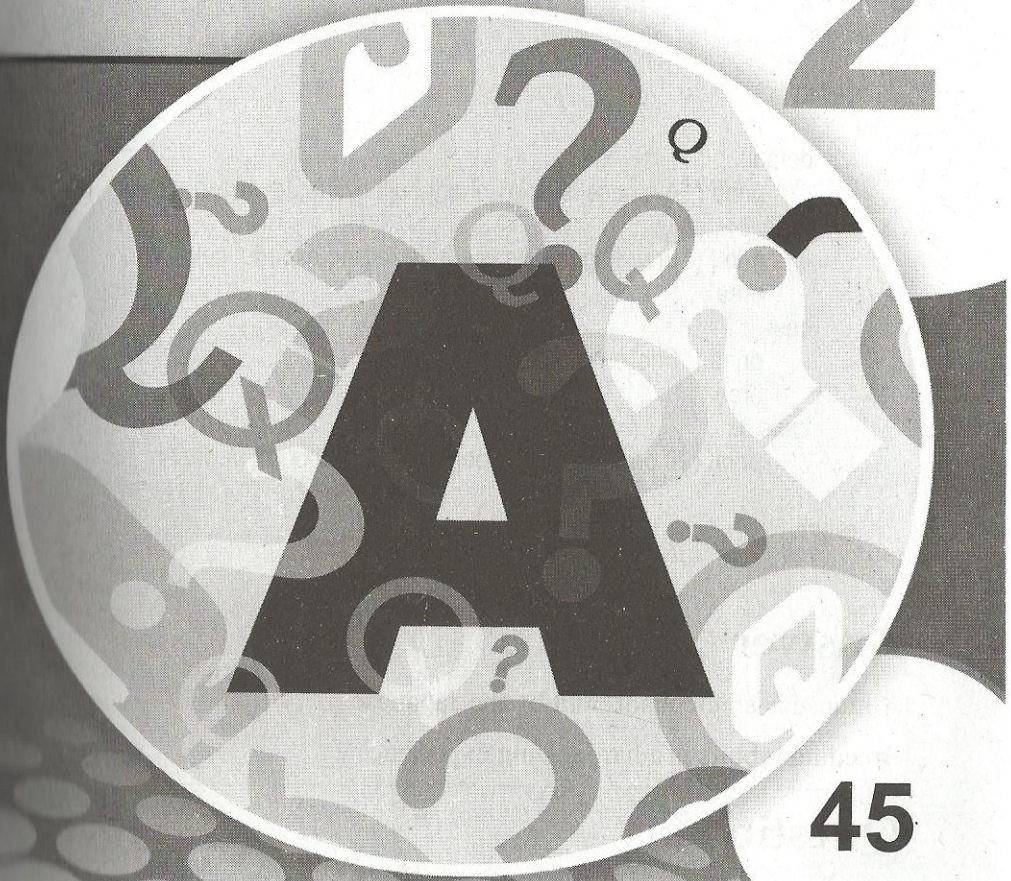
Control Instructions



45

Question 2.1

What will be the output of the following program?

```
#include <stdio.h>
int main( )
{
    int i = 4 ;
    switch ( i )
    {
        default :
            printf ( "A mouse is an elephant built by the Japanese\n" ) ;
        case 1 :
            printf ( "Breeding rabbits is a hare raising experience\n" ) ;
            break ;
        case 2 :
            printf ( "Friction is a drag\n" ) ;
            break ;
        case 3 :
            printf ( "If practice makes perfect, then nobody's perfect\n" ) ;
    }
    return 0 ;
}
```

Answer

A mouse is an elephant built by the Japanese

Breeding rabbits is a hare raising experience

Question 2.2

In the program given below, point out the error, if any, in the *for* loop.

```
#include <stdio.h>
int main( )
{
    int i = 1 ;
    for ( ; ; )
    {
        printf ( "%d\n", i++ ) ;
        if ( i > 10 )
            break ;
    }
    return 0 ;
}
```

A. There should be a condition in the *for* loop.
B. The two semicolons should be dropped.
C. The *for* loop should be replaced by a *while* loop.
D. No error.

Answer

D

Question 2.3

In the program given below, point out the error, if any, in the *while* loop.

```
#include <stdio.h>
int main( )
{
    int i = 1 ;
    while( )
    {
        printf ( "%d\n", i++ ) ;
        if ( i > 10 )
            break ;
    }
    return 0 ;
}
```

A. There should be a condition in the *while* loop.
B. There should be at least a semicolon in the *while( )*.
C. The *while* loop should be replaced by a *for* loop.
D. No error.

Answer

A

Question 2.4

Which of the following statements are correct about an *if-else* statement in a C program?

A. Every *if-else* statement can be replaced by an equivalent statement using *? :* operators.
B. Nested *if-else* statements are allowed.
C. If we use an *if* it is compulsory to use an *else*.
D. Multiple statements in an *if* block are allowed.
E. Multiple statements in an *else* block are allowed.

Answer

B, D, E

Question 2.5

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int x = 10, y = 20 ;
    if ( ! ( !x ) && x )
        printf ( "x = %d\n", x ) ;
    else
        printf ( "y = %d\n", y ) ;
    return 0 ;
}
```

A. y = 20
B. x = 0
C. x = 10
D. x = 1
E. x = -10

Answer

C

Question 2.6

In the program given below, point out the error, if any, in the *while* loop.

```
#include <stdio.h>
int main( )
{
    void fun( ) ;
    int i = 1 ;
    while ( i <= 5 )
    {
        printf ( "%d\n", i ) ;
        if ( i > 2 )
            goto here ;
    }
    return 0 ;
}
void fun( )
{
    here:
```

```
        printf ( "If it works, Don't fix it\n" ) ;
}
```

Answer

goto cannot take control to a function other than the one in which it is being used. In other words, *goto* cannot perform a non-local jump.

Question *2.7*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char i = 0 ;
    for ( i <= 5 && i >= -1 ; ++i ; i > 0 )
        printf ( "%d\n", i ) ;
    printf ( "\n" ) ;
    return 0 ;
}
```

A. 1 2 3 ... 126 127 -128 -127 ... -2 -1
B. Expression syntax error
C. No Output
D. 0 1 2 3 4 5

Answer

A

Question *2.8*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char ch ;
    if ( ( ch = printf ( "" ) ) )
        printf ( "It matters \n" ) ;
    else
        printf ( "It doesn't matter \n" ) ;
    return 0 ;
}
```

A. It matters
B. It doesn't matter
C. matters
D. No Output

Answer

B

Question *2.9*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int i = 1 ;
    switch ( i )
    {
```

```
        printf ( "Hello\n" ) ;
        case 1 :
            printf ( "Hi\n" ) ;
            break ;
        case 2 :
            printf ( "\nBye\n" ) ;
            break ;
    }
    return 0 ;
}
```

A. Hello
Hi

B. Hello
Bye

C. Hi

D. Bye

E. Hello
Hi
Bye

Answer

C

Question *2.10*

Which of the following errors would be reported by the compiler on compiling the program given below?

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



```
{
    int a = 5 ;
    switch ( n )
    {
        case 1 :
          ..
        case 2 :
          ..
        case 3 + 5 :
          ..
        case a :
          ..
    }
    return 0 ;
}
```

A. Variables cannot be checked using a *switch* as in *case a*
B. Expression as in *case 3 + 5* is not allowed
C. All cases in the *switch* are not unique
D. There is no *break* statement in each case
E. There is no *continue* statement in each case

Answer

A, C

Question *2.11*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int a = 500, b = 100, c ;
    if ( !a >= 400 )
        b = 300 ;
    c = 200 ;
```

```
    printf ( "b = %d  c = %d\n", b, c ) ;
    return 0 ;
}
```

A. b = 300 c = 200
B. b = 100 c = garbage
C. b = 300 c = garbage
D. b = 100 c = 200
E. b = 200 c = 300

Answer

D

Question 2.12

Which of the following statements are correct about the C program given below?

```
#include <stdio.h>
int main( )
{
    int x = 10, y = 100 % 90, i ;
    for ( i = 1 ; i <= 10; i++ ) ;
    if ( x != y ) ;
        printf ( "x = %d y = %d\n", x, y ) ;
    return 0 ;
}
```

A. The *printf( )* function is called 10 times.
B. The program will produce the output x = 10 y = 10.
C. The ; after the *if (x!=y)* will NOT produce an error.
D. The program will not produce any output.
E. The *printf( )* function is called infinite times.

Answer

B, C

Question 2.13

Point out the error, if any, in the following program.

```
#include <stdio.h>
int main( )
{
    int i = 4, j = 2 ;
    switch ( i )
    {
        case 1 :
            printf ( "To err is human, to forgive is against  Co. policy\n" ) ;
            break ;
        case j :
            printf ( "If you have nothing to do, don't do it here.\n" ) ;
            break ;
    }
    return 0 ;
}
```

Answer

Constant expression required in the second case, we cannot use *j*.

Question 2.14

Point out the error, if any, in the following program.

```
#include <stdio.h>
int main( )
{
    int i = 1 ;
    switch ( i )
    {
```

```
        case 1 :
            printf ( "Radioactive cats have 18 half-lives.\n" ) ;
            break ;
        case 1 * 2 + 4 :
            printf ( "Bottle for rent - enquire within.\n" ) ;
            break ;
    }
    return 0 ;
}
```

Answer

No error. Constant expressions like 1 * 2 + 4 are acceptable in cases of a *switch*.

Question *2.15*

Point out the error, if any, in the following program.

```
#include <stdio.h>
int main( )
{
    int a = 10 ;
    switch ( a )
    {
    }
    printf ( "Programmers never die. They just get lost in the processing\n" ) ;
    return 0 ;
}
```

Answer

Though never required, there can exist a *switch*, which has no cases.

Question *2.16*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int i = 5 ;
    while ( i-- >= 0 ) printf ( "%d ", i ) ;
    i = 5 ;
    printf ( "\n" ) ;
    while ( i-- >= 0 ) printf ( "%i ", i ) ;
    printf ( "\n" ) ;
    while ( i-- >= 0 ) printf ( "%d ", i ) ;
    printf ( "\n" ) ;
    return 0 ;
}
```

A. 4 3 2 1 0 -1
   4 3 2 1 0 -1

B. 5 4 3 2 1 0
   5 4 3 2 1 0

C. Error

D. 5 4 3 2 1 0
   5 4 3 2 1 0
   5 4 3 2 1 0

Answer

A

Question 2.17

Point out the error, if any, in the following program.

```
#include <stdio.h>
int main( )
{
    int i = 1 ;
    switch ( i )
    {
        printf ( "Hello\n" ) ;  /* common for both cases */
        case 1 :
            printf ( "Individualists unite! \n" ) ;
            break ;
        case 2 :
            printf ( "Money is the root of all wealth.\n" ) ;
            break ;
    }
    return 0 ;
}
```

Answer

Though there is no error, irrespective of the value of *i* the first *printf( )* can never get executed. In other words, all statements in a *switch* have to belong to some *case* or the other.

Question 2.18

Rewrite the following set of statements using conditional operators.

```
int a = 1, b ;
if ( a > 10 )
    b = 20 ;
```


Answer

```
int a = 1, b, dummy ;
a > 10 ? b = 20 : dummy = 1 ;
```

Note that the following would make the first ; work as a statement terminator and the second ; work as a null statement.

```
a > 10 ? b = 20 : ; ;
```

Question 2.19

Point out the error, if any, in the following program.

```
#include <stdio.h>
int main( )
{
    int a = 10, b ;
    a >= 5 ? b = 100 : b = 200 ;
    printf ( "%d\n", b ) ;
    return 0 ;
}
```

Answer

Visual Studio does not report any error for this program. However, Turbo C/C++ compiler objects saying 'Lvalue required in function *main( )*'. To avoid this error the second assignment should be written in parentheses as follows:

```
a >= 5 ? b = 100 : ( b = 200 ) ;
```

Question 2.20

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int i = 0 ;
    while ( i++ != 0 )
        printf ( "%d ", ++i ) ;
    printf ( "\n" ) ;
    return 0 ;
}
```

A. 0 1 2 127
B. 0 1 2 65535
C. 1 2 .... 32767 -32766 -32765 .... -1 0
D. No output

Answer

D

Question 2.21

Which of the following statements is correct about the program given below?

```
#include <stdio.h>
int main( )
{
    int i = 0 ;
    for ( i = 0 ; i <= 127 ; printf ( "%d", i++ ) )
        ;
    printf ( "\n" ) ;
    return 0 ;
}
```


A. The program would go in an infinite loop
B. The program would output 0 1 2 .... 126 127
C. The program would not produce any output
D. The program would report an error: Cannot use printf( ) in for loop

Answer

B

Question 2.22

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char str[ ] = "Part-time musicians are semiconductors" ;
    int a = 5 ;
    printf ( a > 10 ? "%50s\n" : "%s\n", str ) ;
    return 0 ;
}
```

A. Part-time musicians are semiconductors
B. Part-time musicians are semiconductors
C. Error
D. None of the above

Answer

A

Question 2.23

What is more efficient a *switch* statement or an *if-else* chain?

Answer

As far as efficiency is concerned there would hardly be any difference, if at all. If the cases in a *switch* are sparsely distributed the compiler may internally use the equivalent of an *if-else* chain instead of a compact jump table. However, one should use *switch* where one can. It is definitely a cleaner way to program and certainly is not any less efficient than the *if-else* chain.

Question 2.24

Can we use a *switch* statement to switch on strings?

Answer

No. The cases in a *switch* must either have integer constants or constant expressions.

Question 2.25

We want to test whether a value lies in the range 2 to 4 or 5 to 7. Can we do this using a switch?

Answer

Yes, though in a way which would not be very practical if the ranges are bigger. The way is shown below:

```
switch ( a )
{
    case 2 :
    case 3 :
    case 4 :
        /* some statements */
        break ;
    case 5 :
    case 6 :
    case 7 :
        /* some other statements */
        break ;
}
```

Question 2.26

The way *break* is used to take the control out of *switch* can *continue* be used to take the control to the beginning of the *switch*? [Yes/No]

Answer

No. *continue* can work only with loops and not with *switch*.

Question 2.27

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int a = 300, b, c ;
    if ( a >= 400 )
        b = 300 ;
        c = 200 ;
    printf ( "%d %d %d\n", a, b, c ) ;
    return 0 ;
}
```

A. 300 300 200
B. Garbage 300 200
C. 300 Garbage 200
D. 300 300 Garbage

Answer

C

Question 2.28

Which of the following statements are correct about the program given below?

```
#include <stdio.h>
int main( )
{
    int x = 30, y = 40 ;
    if ( x == y )
        printf ( "x is equal to y\n" ) ;
    elseif ( x > y )
        printf ( "x is greater than y\n" ) ;
    elseif ( x < y )
        printf ( "x is less than y\n" ) ;
    return 0 ;
}
```

A. Error: '*elseif* is not a keyword in C'
B. Error: 'Expression syntax'
C. Error: 'Lvalue required'
D. Error: 'Rvalue required'

Answer

A

To make the program work replace 'elseif' with 'else if'.

Question 2.29

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    float a = 0.7 ;
    if ( 0.7 > a )
        printf ( "Hi\n" ) ;
    else
        printf ( "Hello\n" ) ;
    return 0 ;
}
```

A. Hi
B. Hello
C. Ho Hello
D. None of the above

Answer

A.

Here *a* is a *float*, whereas 0.7 is a *double*. When 0.7 is stored in *a* what get stored is a 32-bit binary equivalent of 0.7. The binary equivalent of 0.7 turns out to be a recurring number. This recurring number is terminated at 32 bits when it is stored in *a*. As against this, when 0.7 is treated as a *double* the recurring binary equivalent of 0.7 is terminated at 64 bits. When the 64-bit recurring binary equivalent of 0.7 is compared with 32-bit recurring binary equivalent of 0.7, the 64-bit value turns out to be greater than the 32-bit value. Hence the condition turns out to be true.

Question 2.30

How would you convert the following program to eliminate the *if.. else if .. else* clause?

```
#include <stdio.h>
int main( )
{
    int x ;
    printf ( "Enter value of x as 0 or 1\n" ) ;
    scanf ( "%d", &x ) ;
    if ( x == 0 )
        printf ( "x is equal to zero\n" ) ;
    else if ( x == 1 )
        printf ( "x is equal to one\n" ) ;
    else
        printf ( "You entered a wrong number\n" ) ;
    return 0 ;
}
```

Answer

```
#include <stdio.h>
int main( )
{
    int x ;
    printf ( "Enter value of x as 0 or 1\n" ) ;
    scanf ( "%d", &x ) ;
    switch ( x )
    {
        case 0 :
            printf ( "x is equal to zero\n" ) ;
            break ;
        case 1 :
            printf ( "x is equal to one\n" ) ;
            break ;
        default :
            printf ( "You entered a wrong number\n" ) ;
        return 0 ;
    }
}
```


Question 2.31

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int i = 3 ;
    switch ( i )
    {
        case 1 :
            printf ( "Hello\n" ) ;
        case 2 :
            printf ( "Hi\n" ) ;
            break ;
        case 3 :
            continue ;
        default :
            printf ( "Bye\n" ) ;
    }
    return 0 ;
}
```

A. Error :'Misplaced Continue'
B. Bye
C. No output
D. Hello Hi

Answer

A

Question 2.32

Which of the following statements are correct about the program given below?

```
#include <stdio.h>
int main( )
{
    char x ;
    while ( x = 0 ; x <= 255 ; x++ )
        printf ( "Ascii value %d Character %c\n", x, x ) ;
    return 0 ;
}
```

A. The program goes in an infinite loop.
B. The program prints all the ASCII values with its corresponding characters.
C. Error: x should be declared as an *int*.
D. The program reports an error as *while* loop cannot take the form of a *for* loop.

Answer

D

Question 2.33

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int k, num = 30 ;
    k = ( num > 5 ? ( num <= 10 ? 100 : 200 ) : 500 ) ;
    printf ( "%d\n", k ) ;
    return 0 ;
}
```

A. 200
B. 5
C. 100
D. 500

Answer

A

Question 2.34

Which of the following is not a logical operator?

A. &
B. &&
C. ||
D. !

Answer

A

Question 2.35

Which of the following cannot be checked in a *switch – case* statement?

A. Character
B. Integer
C. Float

D. enum

Answer

C

Question 2.36

Which of the following statements are correct about a *for* loop used in a C program?

A. *for* loop works faster than a *while* loop.
B. Everything that can be done using a *for* loop can also be done using a *while* loop.
C. *for (;;)* implements an infinite loop.
D. *for* loop can be used if we want statements in a loop to get executed at least once.
E. *for* loop works faster than a *do-while* loop.

Answer

B, C, D

Question 2.37

Which of the following statements are correct about the program given below?

```
#include <stdio.h>
int main( )
{
    int i = 10, j = 20 ;
    if ( i == 5 ) && if ( j == 10 )
        printf ( "Have a nice day\n" ) ;
    return 0 ;
}
```


A. Output: Have a nice day
B. No output
C. Error: 'Expression syntax'
D. Error: 'Undeclared identifier if'

Answer

C.

The correct form of *if* would be:

```
if ( ( i == 5 ) && ( j == 10 ) )
    printf ( "Have a nice day\n" ) ;
```

Question 2.38

Which of the following statements are correct about the program given below?

```
#include <stdio.h>
int main( )
{
    int n = 0, y = 1 ;
    y == 1 ? ( n = 0 ) : ( n = 1 ) ;
    if ( n )
        printf ( "YES\n" ) ;
    else
        printf ( "NO\n" ) ;
    return 0 ;
}
```

A. Error: 'Declaration terminated incorrectly'
B. Error: 'Syntax error'
C. Error: 'Lvalue required'
D. Error: 'Expression syntax'

E. The program works correctly without error.

Answer

E

Question 2.39

Which of the following statements are correct about a *switch* statement in a C program?

A. *switch* is useful when we wish to check the value of a variable against a particular set of values.
B. *switch* is useful when we wish to check whether a value falls in different ranges.
C. Compiler implements a jump table for cases used in a *switch*.
D. It is NOT compulsory to use a *break* in every *switch* statement.
E. A common set of statements CANNOT be used for multiple *cases* in a *switch*.

Answer

A, C, D

Question 2.40

Which of the following statements are correct about the program given below?

```
#include <stdio.h>
#include <stdlib.h>
int main( )
{
    int i = 0 ;
    i++ ;
```



```
    if ( i <= 5 )
    {
        printf ( "adds wings to your thoughts\n" ) ;
        exit ( 0 ) ;
        main( ) ;
    }
    return 0 ;
}
```

A. The program prints 'adds wings to your thoughts' five times.
B. The program prints 'adds wings to your thoughts' infinite times.
C. The program prints 'adds wings to your thoughts' once.
D. The call to *main( )* after *exit (0)* doesn't materialize.
E. The compiler reports an error since *main( )* cannot call itself.

Answer

C, D

Question 2.41

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int x = 3 ;
    float y = 3.0 ;
    if ( x == y )
        printf ( "x and y are equal\n" ) ;
    else
        printf ( "x and y are not equal\n" ) ;
    return 0 ;
}
```

A. x and y are equal
B. x and y are not equal
C. Unpredictable
D. No output

Answer

A

During comparison *x* would get promoted to a *float* and then two *float*s would be compared.

Question 2.42

Which of the following statements are correct about the following program?

```
#include <stdio.h>
int main( )
{
    int i = 10, j = 15 ;
    if ( i % 2 = j % 3 )
        printf ( "\nCarpathians" ) ;
    return 0 ;
}
```

A. Error: 'Expression syntax'
B. Error: 'Lvalue required'
C. Error: 'Rvalue required'
D. The code runs successfully.

Answer

B

Question 2.43

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int a = 0, b = 1, c = 3 ;
    * ( ( a ) ? &b : &a ) = a ? b : c ;
    printf ( "%d %d %d\n", a, b, c ) ;
    return 0 ;
}
```

A. 0 1 3
B. 1 2 3
C. 3 1 3
D. 1 3 1

Answer

C

Question 2.44

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int i = 0 ;
    for ( ; i < 5 ; i++ ) ;
        printf ( "%d \n" , i ) ;
    return 0 ;
}
```

A. 0 1 2 3 4
B. 5
C. 1 2 3 4
D. 6

Answer

B

Question 2.45

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char j = 1 ;
    while ( j <= 255 )
    {
        printf ( "%d ", j ) ;
        j = j + 1 ;
    }
    printf ( "\n" ) ;
    return 0 ;
}
```

A. 1 2 3 .. 127
B. 1 2 3 .. 255
C. 1 2 3 .. 254 255 0 1 2 3 .. 254 255 .. infinite times
D. 1 2 3 .. 127 128 0 1 2 3 .. 127 128 .. infinite times
E. 1 2 3 .. 127 -128 -127 -126 .. -2 -1 0 1 2 .. 127 -128 -127 .. infinite times

Answer

E

Question 2.46

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int x = 1, y = 1 ;
    for( ; y ; printf ( "%d %d\n", x, y ) )
        y = x++ <= 5 ;
    return 0 ;
}
```

A. 2 1
   3 1
   4 1
   5 1
   6 1
   7 0

B. 2 1
   3 1
   4 1
   5 1
   6 1

C. 2 1
   3 1
   4 1
   5 1

D. 2 1

3 1
4 1
4 1

E. 2 2
3 3
4 4
5 5

Answer

A

Question 2.47

On executing the following program how many times will the message "Keep it up" get printed?

```
#include <stdio.h>
int main( )
{
    int x ;
    for ( x = -1; x <= 10 ; x++ )
    {
        if ( x < 5 )
            continue ; → it will not print
        else
            break ;
        printf ( "Keep it up\n" ) ;
    }
    return 0 ;
}
```

A. Infinite times
B. 11 times
C. 0 times

D. Once
E. 10 times

Answer

C

Question 2.48

How many times the *while* loop in the following program will get executed if a *short int* is 2 byte wide?

```
#include <stdio.h>
int main( )
{
    int j = 1 ;
    while ( j <= 255 ) ; → semicolon
    {
        printf ( "%c %d \n", j, j ) ;
        j++ ; → it will never come to j++
    }
    return 0 ;
}
```

A. Infinite times
B. 255 times
C. 256 times
D. Only once
E. 254 times

Answer

A

Question 2.49

How would you rewrite the following program using a *while* and a *for* loop?

```
#include <stdio.h>
int main( )
{
    char ch = 'Y' ;
    do
    {
        /* some logic */
        printf ( "Continue Y/N " ) ;
        fflush ( stdin ) ;
        scanf ( "%c", &ch ) ;
    } while ( ch == 'Y' ) ;
}
```

Answer

Program using *for* loop:

```
#include <stdio.h>
int main( )
{
    char ch = 'Y' ;
    for ( ; ; )
    {
        /* some logic */
        printf ( "Continue Y/N " ) ;
        fflush ( stdin ) ;
        scanf ( "%c", &ch ) ;
        if ( ch != 'Y' )
            break ;
    }
}
```


Program using *while* loop:

```
#include <stdio.h>
int main( )
{
    char ch = 'Y' ;
    while ( 1 )
    {
        /* some logic */
        printf ( "Continue Y/N " ) ;
        fflush ( stdin ) ;
        scanf ( "%c", &ch ) ;
        if ( ch != 'Y' )
            break ;
    }
}
```