1. For 16-bit compiler allowable range for integer constants is \_\_\_\_\_\_?
   1. -3.4e38 to 3.4e38
   2. -32767 to 32768
   3. -32768 to 32767
   4. -32668 to 32667

Answer: C

1. The statement printf ("%d", 10? 0? 5: 1: 12); will print?
2. 10
3. 0
4. 12
5. 1

Answer: D

1. Predict the output of following program?

void main()

{

int a=10,b=20;

char x=1,y=0;

if(a, b, x, y)

{

printf("EXAM");

}

}

What is the output?

1. XAM is printed
2. exam is printed
3. Compiler Error
4. Nothing is printed

Answer : D

1. What will be the value of `a` after the following code is executed

#define square(x) x\*x

a = square(2+3)

1. 25
2. 13
3. 11
4. 10

Ans: c

1. Predict the output of following program?

void main()

{

int a,b;

a=1, 3, 15;

b= (2, 4, 6);

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

}

1. 3
2. 21
3. 17
4. 7
5. Compiler error

Answer: D

1. Predict the output of following program?

void main()

{

if(printf("cquestion"))

printf("I know c");

else

printf("I know c++");

}

1. I know c
2. I know c++
3. cquestionI know c
4. cquestionI know c++

Answer : C

1. continue statement is used
2. to go to the next iteration in a loop
3. come out of a loop
4. exit and return to the main function
5. Restarts iterations from beginning of loop.

Answer : D

1. Predict the output of following program?

#include<stdio.h>

int main(){

int a;

a=015 + 0x71 +5;

printf("%d",a);

return 0;

}

1. 130
2. 256
3. 131
4. 132

Output: 131

Explanation:

015 is octal number its decimal equivalent is = 5 \* 8 ^ 0 + 1 \* 8 ^ 1 = 5 + 8 = 13

0x71 is hexadecimal number (0x is symbol of hexadecimal) its decimal equivalent is = 1 \* 16 ^ 0 + 7 \* 16 ^ 1 = 1 + 112 = 113

So, a = 13 + 113 + 5 = 131

1. Predict the output of following program?

#include<stdio.h>

int main(){

int a=2;

a=a++ + ~++a;

printf("%d",a);

return 0;

}

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

Answer : B

1. Predict the output of following program?

#include<stdio.h>

int main(){

int i=5;

int a=++i + ++i + ++i;

printf("%d",a);

return 0;

}

1. 21
2. 15
3. 24
4. 18

Answer : A

Explanation:

Rule : ++ (in ++i) is pre increment operator so in any arithmetic expression it first increment the value of variable by one in whole equation up to break point then start assigning the value of variable in the equation. There are many break point operators in. For example:

(1) Declaration statement.

(2) && or operator.

(3) Comma (,) operator etc.

In the following expression:

int a=++i + ++i + ++i;

Here break point is due to declaration .It break after each increment i.e. (initial value of i=5) after first increment value 6 assign to variable i then in next increment will occur and so on.

So, a = 6 + 7 + 8;

1. Predict the output of following program?

#include "stdio.h"

int main()

{

char arr[10];

printf("%d", scanf("%s", arr));

/\* Suppose that input value given for above scanf is "JIET" \*/

return 1;

}

1. 9
2. 1
3. 10
4. 100

Answer : B

Explanation: In C, scanf returns the no. of inputs it has successfully read.

1. What does the following C statement mean?

scanf("%4s", str);

Run on IDE and find the output

1. Read exactly 4 characters from console.
2. Read maximum 4 characters from console.
3. Read a string str in multiples of 4
4. Nothing

Answer : B

1. Find Output of following program?

#include<stdio.h>

int main()

{

printf("%d", printf("%d", 1234));

return 0;

}

1. 12344
2. 12341
3. 11234
4. 41234

Answer : A

1. What is the return type of getchar()?
2. int
3. char
4. unsigned char
5. float

Answer : A

Explanation: The return type of getchar() is int to accommodate EOF which indicates failure.

1. Predict the output of following program?

#include <stdio.h>

int main()

{

int i = 3;

printf("%d", (++i)++);

return 0;

}

What is the output of the above program?

1. 3
2. 4
3. 5
4. Compile-time error

Answer : D

Explanation:

In C, prefix and postfix operators need l-value to perform operation and return r-value. The expression (++i)++ when executed increments the value of variable i(i is a l-value) and returns r-value. The compiler generates the error(l-value required) when it tries to post-incremeny the value of a r-value.

1. Predict the output of following program?

#include <stdio.h>

int main()

{

//Assume sizeof character is 1 byte and sizeof integer is 4 bytes

printf("%d", sizeof(printf("JIET UNIVESE")));

return 0;

}

1. JIET UNIVESE12
2. JIET UNIVESE4
3. 4
4. Compile-time error

Answer : C

Explanation:

An expression doesn't get evaluated inside sizeof operator. GeeksQuiz will not be printed. printf returns the number of characters to be printed i.e. 9 which is an integer value. sizeof operator returns sizeof(int).

1. Predict the output of the below program:

#include <stdio.h>

int main()

{

int a = 1;

int b = 1;

int c = a || --b;

int d = a-- && --b;

printf("a = %d, b = %d, c = %d, d = %d", a, b, c, d);

return 0;

}

1. a = 0, b = 1, c = 1, d = 0
2. a = 0, b = 0, c = 1, d = 0
3. a = 1, b = 1, c = 1, d = 1
4. a = 0, b = 0, c = 0, d = 0

Answer : B

Explanation:

Let us understand the execution line by line. Initial values of a and b are 1.

// Since a is 1, the expression --b is not executed because

// of the short-circuit property of logical or operator

// So c becomes 1, a and b remain 1

int c = a || --b;

// The post decrement operator -- returns the old value in current expression

// and then updates the value. So the value of expression --a is 1. Since the

// first operand of logical and is 1, shortcircuiting doesn't happen here. So

// the expression --b is executed and --b returns 0 because it is pre-increment.

// The values of a and b become 0, and the value of d also becomes 0.

int d = a-- && --b;

1. Predict the output of the below program:

#include <stdio.h>

int main()

{

printf("%d", 1 << 2 + 3 << 4);

return 0;

}

1. 112
2. 52
3. 512
4. 0

Answer : C

Explanation:

The main logic behind the program is the precedence and associativity of the operators. The addition(+) operator has higher precedence than shift(<<) operator. So, the expression boils down to 1 << (2 + 3) << 4 which in turn reduces to (1 << 5) << 4 as the shift operator has left-to-right associativity.

1. Predict the output of following program?

#include<stdio.h>

int main()

{

int a = 2,b = 5;

a = a^b;

b = b^a;

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

return 0;

}

1. 5 2
2. 2 5
3. 7 7
4. 7 2

Answer : D

Explanation:

^ is bitwise xor operator. a = 2 (10) b = 5 (101) a = a^b (10 ^ 101) = 7(111) b = a^b (111 ^ 101) = 2(10)

1. Predict the output of following program?

# include <stdio.h>

int main()

{

int x = 10;

int y = 20;

x += y += 10;

printf (" %d %d", x, y);

return 0;

}

1. 40 20
2. 40 30
3. 30 30
4. 30 40

Answer : B

Explanation:

The main statement in question is "x += y += 10". Since there are two += operators in the statement, associativity comes into the picture. Associativity of compound assignment operators is right to left, so the expression is evaluated as x += (y += 10).

1. Predict the output of following program?

#include <stdio.h>

int main()

{

int x = 10;

int y = (x++, x++, x++);

printf("%d %dn", x, y);

return 0;

}

1. 13 12
2. 13 13
3. 10 10
4. Compiler Dependent

Answer : A

Explanation:

The comma operator defines a sequence point, so the option (d) is not correct. All expressions are executed from left to right and the value of rightmost expression is returned by the comma operator.

1. Predict the output of following program?

#include <stdio.h>

int main()

{

int y = 0;

int x = (~y == 1);

printf("%d", x);

return 0;

}

1. 0
2. 1
3. negative Number
4. Compiler Error

Answer : A

Explanation:

The important thing to note here is ~ is a bitwise not operator. So the value of ~0 would be all 1s in binary representation which means decimal value of ~0 is not 1. Therefore the result of comparison operator becomes 0.

1. Predict the output of following program?

#include<stdio.h>

int main(){

int i,j,k;

for(i=0,j=2,k=1;i<=4;i++){

printf("%d ",i+j+k);

}

return 0;

}

1. 3 4 5 6 7
2. 3 5 7 9 10
3. 3 5 6 7
4. 3 4 5 6 7 8

Answer : A

1. Predict the output of following program?

void main(){

int i,j=2;

for(i=0;i<=5,j>=0;i++){

printf("%d ",i+j);

j--;

}

1. 2 2 2
2. 1 1 1
3. 3 3 3
4. 1 2 3

Answer : A

1. Predict the output of following program?

#include<stdio.h>

int main(){

int i,j,k;

for(i=0,j=0,k=0;i<=5,j<=4,k<=3;i++,++j,k+=2)

printf("%d ",i+j+k);

}

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

Answer : A

1. What will be output when you will execute following c code?

#include<stdio.h>

void main(){

switch(5||2|1){

case 3&2:printf("C");

break;

case -~11:printf("C++");

break;

case 6-3<<2:printf("JAVA");

break;

case 5>=5:printf("C#");

}

}

1. C
2. JAVA
3. C++
4. Compilation error

Answer : D

Consider on the expression:

5||2|1

=5|| (2|1) //Bitwise or has higher precedence

=5||3

=1

Now, value of each case expression:

3&2 = 2

-~11 = -(-12) =12

6-3<<2 = 3 <<2 = 12

5>=5 = 1

case -~11 and case 6-3<<2 have same constant expression i.e. case 12

In c duplicate case is not possible.

1. What will be output when you will execute following c code?

#include<stdio.h>

void main(){

switch(6){

case 6.0f:printf("C");

break;

case 6.0: printf("C++");

break;

case 6.0L:printf("JAVA");

break;

default: printf(“C#");

}

}

1. C
2. C++
3. JAVA
4. Compilation error

Answer : D

Explanation:

Case expression must be integral constant expression. If it is not integer then it is automatically type casted into integer value.

so. (int)6.0f = 6

(int)6.0 = 6

(int)6.0L = 6

In c duplicate case is not possible.

1. What will be output when you will execute following c code?

#include<stdio.h>

void main(){

switch(5/2\*6+3.0){

case 3:printf("C ");

break;

case 15:printf("C++");

break;

case 0:printf("JAVA");

break;

default:printf("C#");

}

}

1. C
2. C++
3. JAVA
4. Compilation error Answer : D

Consider on the expression:

5/2\*6+3.0

=2\*6+3.0

=12 + 3.0

=15.0

In c switch expression must return an integer value. It cannot be float, double or long double

1. #include<stdio.h>

int main()

{

char c = 125;

c = c+10;

printf("%d", c);

return 0;

}

1. 135
2. +INF
3. -121
4. -8

Answer : C

1. #include <stdio.h>

int main()

{

if (sizeof(int) > -1)

printf("Yes");

else

printf("No");

return 0;

}

1. Yes
2. No
3. Compiler Error
4. Runtime Error

Answer : B