**Argument Passing**

**Writer: Sayak Haldar**

**1. How many ways of passing a parameter are there in c++?**

a) 1

b) 2

c) 3

d) 4

View Answer

**Answer: c**

Explanation: There are three ways of passing a parameter. They are pass by value,pass by reference and pass by pointer.

**2. Which is used to keep the call by reference value as intact?**

a) static

b) const

c) absolute

d) none of the mentioned

**Answer: b**

Explanation: Because const will not change the value of the variables during the execution.

**3. By default how the value are passed in c++?**

a) call by value

b) call by reference

c) call by pointer

d) none of the mentioned

**Answer: a**

**4. What is the output of this program?**

#include <iostream>

using namespace std;

void copy (int& a, int& b, int& c)

{

a \*= 2;

b \*= 2;

c \*= 2;

}

int main ()

{

int x = 1, y = 3, z = 7;

copy (x, y, z);

cout << "x =" << x << ", y =" << y << ", z =" << z;

return 0;

}

a) 2 5 10

b) 2 4 5

c) 2 6 14

d) none of the mentioned

View Answer

**Answer: c**

Explanation: Because we multiplied the values by 2 in the copy function.

Output:

$ g++ arg6.cpp

$ a.out

x = 2,y = 6,z = 14

**5. What is the new value of x?**

#include <iostream>

using namespace std;

void fun(int &x)

{

x = 20;

}

int main()

{

int x = 10;

fun(x);

cout << "New value of x is " << x;

return 0;

}

a) 10

b) 20

c) 15

d) none of the mentioned

**Answer: b**

Explanation: As we passed by reference, the value is changed and it is returned as 20.

Output:

$ g++ arg5.cpp

$ a.out

20

**6. What is the output of this program?**

#include <iostream>

using namespace std;

long factorial (long a)

{

if (a > 1)

return (a \* factorial (a + 1));

else

return (1);

}

int main ()

{

long num = 3;

cout << num << "! = " << factorial ( num );

return 0;

}

a) 6

b) 24

c) segmentation fault

d) compile time error

**Answer: c**

Explanation: As we have given in the function as a+1, it will exceed the size and so it arises the segmentation fault.

Output:

$ g++ arg3.cpp

$ a.out

segmentation fault

**7. What is the output of this program?**

#include <iostream>

using namespace std;

void square (int \*x)

{

\*x = (\*x + 1) \* (\*x);

}

int main ( )

{

int num = 10;

square(&num);

cout << num;

return 0;

}

a) 100

b) compile time error

c) 144

d) 110

**Answer: d**

Explanation: We have increased the x value in operand as x+1, so it will return as 110.

Output:

$ g++ arg2.cpp

$ a.out

110

**8. What is the output of this program?**

#include <iostream>

using namespace std;

int add(int a, int b);

int main()

{

int i = 5, j = 6;

cout << add(i, j) << endl;

return 0;

}

int add(int a, int b )

{

int sum = a + b;

a = 7;

return a + b;

}

a) 11

b) 12

c) 13

d) compile time error

**Answer) c) 13**

**9. What will happen when we use void in argument passing?**

a) It will not return value to its caller

b) It will return value to its caller

c) Maybe or may not be return any value to its caller

d) None of the mentioned

**Answer: a**

Explanation: As void is not having any return value, it will not return the value to the caller.

**10. What is the output of this program?**

#include <iostream>

using namespace std;

void Sum(int a, int b, int & c)

{

a = b + c;

b = a + c;

c = a + b;

}

int main()

{

int x = 2, y =3;

Sum(x, y, y);

cout << x << " " << y;

return 0;

}

a) 2 3

b) 6 9

c) 2 15

d) compile time error

**Answer: c**

Explanation: We have passed three values and it will manipulate according to the given condition and yield the result as 2 15

Output:

$ g++ arg.cpp

$ a.out

2 15