

Programming in Modern C++: Assignment Week 1

Total Marks : 25

Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, Kharagpur – 721302
partha.p.das@gmail.com

January 12, 2023

Question 1

Consider the following program.

[MCQ, Marks 2]

```
#include<iostream>
#include<algorithm>
#include<string.h>
using namespace std;

bool compare(char c1, char c2){
    return tolower(c1) > tolower(c2);    //LINE-1
}

int main(){
    char arr1[20] = "C++ Program", arr2[20] = "C Program";
    cout << lexicographical_compare(arr1, arr1+strlen(arr1), arr2, arr2+strlen(arr2),
                                    compare);
    return 0;
}
```

What will be the output/error?

- a) 1
- b) 0
- c) -1
- d) Compilation Error: function is not defined

Answer: a)

Explanation:

As per the LINE-1, it will return 1 if the first string is lexicographically larger than the second string which is TRUE in our case. Hence the program will print 1.

Question 2

Consider the following code segment.

[MSQ, Mark 1]

```
#include <iostream>
#include <algorithm>
using namespace std;
int main() {
    int data[] = {1,2,3,4,5};
    int key = 5;
    if(binary_search(____))    //LINE-1
        cout << "found";
    else
        cout << "not found";
    return 0;
}
```

Fill in the blank at LINE-1 so that the program will print "not found"?

- a) &data[0], &data[5], key
- b) data, data+5, key
- c) &data[0], &data[4], key
- d) data+1, data+4, key

Answer: c), d)

Explanation:

`binary_search(.)` function takes 3 parameters. The first two parameters are starting and ending addresses of the array where you want to search, and the third parameter is the key to search. Hence, the correct option is c) and d).

Question 3

Consider the following code segment.

[MCQ, Marks 2]

```
#include <iostream>
#include <algorithm>
using namespace std;
int main () {
    int data[] = {50, 30, 40, 10, 20};
    sort (&data[1], &data[4]);
    for (int i = 0; i < 5; i++)
        cout << data[i] << " ";
    return 0;
}
```

What will be the output?

- a) 10 20 30 40 50
- b) 10 30 40 50 20
- c) 50 10 30 40 20
- d) 50 10 20 30 40

Answer: c)

Explanation:

Since the call is `sort(&data[1], &data[4])`, it considers 3 elements of the array `data[]` from the second element for sorting. Thus, it prints 50 10 30 40 20.

Question 4

Consider the following code segment.

[MCQ, Marks 2]

```
#include <iostream>
#include <algorithm>
using namespace std;
int main() {
    int element[5];
    for(int i = 1; i <= 5; i++)
        *(element + i - 1) = i * 5;
    rotate(element, element + 4, element + 5);
    rotate(element, element + 1, element + 4);
    for (int i = 0; i < 5; ++i)
        cout << element[i] << " ";
    return 0;
}
```

What will be the output?

- a) 5 10 15 20 25
- b) 5 10 15 25 20
- c) 20 10 15 25 5
- d) 25 5 10 15 20

Answer: b)

Explanation:

`rotate(first, middle, last)` rotates the order of the elements in the range `[first,last)`, in such a way that the element pointed by `middle` becomes the new first element.

`rotate(element, element + 4, element + 5)` makes the order as 25 5 10 15 20.

`rotate(element, element + 1, element + 4)` makes the order as 5 10 15 25 20.

Question 5

Consider the following code segment.

[MCQ, Marks 2]

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    const int size = 3, c = 65;
    vector<char> vc(size, 'A');
    for (int i = 1; i <= 2; i++)
        vc.push_back(65 + i);
    vc.resize(10, 90);
    vc.resize(8);
    for (int i = 0; i < vc.size(); i++)
        cout << vc[i] << " ";
    return 0;
}
```

What will be the output?

- a) A A A B C Z Z Z
- b) A A B B C Z Z Z
- c) A A A B C Z Z
- d) A A A B C Z Z Z Z

Answer: a)

Explanation:

Vectors are similar to dynamic arrays having the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container.

The statements and the states of the vector are as follows:

```
vector<char> vc(size, 'A'); ) ['A', 'A', 'A'],
vc.push_back(65 + i); ) ['A', 'A', 'A', 'B', 'C'] (please note the 65 is the ASCII
value of 'A'),
vc.resize(10, 90); ) ['A', 'A', 'A', 'B', 'C', 'Z', 'Z', 'Z', 'Z', 'Z'],
vc.resize(8); ) ['A', 'A', 'A', 'B', 'C', 'Z', 'Z', 'Z'].
```

Question 6

Consider the following code segment.

[MSQ, Marks 2]

```
#include <iostream>
#include <string>
using namespace std;

int main(void) {
    string s1 = "C++ ";
    string s2 = "Program";
    -----;    //LINE-1
    cout << s1;
    return 0;
}
```

Choose the appropriate option to fill in the blank at LINE-1, such that the output of the code would be: C++ Program.

- a) `s1 += s2`
- b) `strcat(s1, s2)`
- c) `s1.append(s2)`
- d) `s1.insert(s2)`

Answer: a), c)

Explanation:

In C++, `operator+=` and `append(·)` append the strings. Please note that `strcat(·)` is a C function, and required the inclusion of `cstring`. The function `insert(·)` is used to insert a string in another string at a given position.

Question 7

Consider the following code segment.

[MCQ, Marks 2]

```
#include <iostream>
#include <algorithm>
using namespace std;
int main() {
    int data[] = { 1, 2, 3, 4, 5 };
    for (int i = 0; i < 1; i++) {
        int j = data[i];
        replace(data, data + 5, j, *(_____));    //LINE-1
    }
    for (int i = 0; i < 5; ++i)
        cout << data[i] << " ";
    return 0;
}
```

Fill in the blank at LINE-1 such that the output is
5 2 3 4 5

- a) data + 4 - i
- b) data + 5 - i
- c) data + i - 4
- d) data + i - 5

Answer: a)

Explanation:

The statement: `replace(data, data + 5, j, *(data + 4 - i));` swap the left-most element with the right-most element and modify the starting position of the array. Hence, LINE-1 will be filled as `data + 4 - i`.

Question 8

Consider the following code segment.

[MCQ, Marks 2]

```
#include <iostream>
#include <cstring>
#include <stack>

using namespace std;

int main(){
    char str[10] = "123456789";
    stack<char> s1, s2;
    int i;
    for(i = 0; i < strlen(str)/2; i++)
        s1.push(str[i]);
    for(i=i-1; i < strlen(str); i++)
        s2.push(str[i]);

    while (!s1.empty()) {
        s2.push(s1.top()); s1.pop();
    }
    while (!s2.empty()) {
        cout << s2.top(); s2.pop();
    }
    return 0;
}
```

What will be the output?

- a) 1234987654
- b) 123498765
- c) 1234897654
- d) 123459876

Answer: a)

Explanation:

The stack `s1` stores {1, 2, 3, 4} and the stack `s2` stores {9, 8, 7, 6, 5, 4}. Then the elements of `s1` are also pushed into `s2` as {1, 2, 3, 4, 9, 8, 7, 6, 5, 4}. Thus, when we finally pop and print the elements from `s2`, the output would be 1234987654.

Question 9

Consider the following code segment.

[MCQ, Marks 2]

```
int i = 5;
const int *p = &i;
int * const q = &i;
int const *r = &i;
int const * const s = &i;

*p = 10; //STMT-1
*q = 10; //STMT-2
*r = 10; //STMT-3
*s = 10; //STMT-4
```

Which statement/statements is/are correct?

- a) STMT-1
- b) STMT-2
- c) STMT-3
- d) STMT-4

Answer: b)

Explanation:

In statement `const int *p = &i;`, for p the pointee is constant, hence `*p` cannot be modified. So a) is incorrect.

In statement `int const *r = &i;`, again for r the pointee is constant, hence `*r` cannot be modified. So c) is incorrect.

In statement `int const * const s = &i;`, for s the pointer and pointee both are constant, hence `*s` cannot be modified. So d) is incorrect.

But in statement `int const *q = &i;`, for q the pointer is constant, hence `*q` can be modified. So b) is the correct option.

Programming Questions

Question 1

Consider the program below.

- Fill in the blank at LINE-1 to declare a stack variable **s**.
- Fill in the blank at LINE-2 to push value into stack.
- Fill in the blank at LINE-3 with appropriate statement.

The program must satisfy the given test cases.

Marks: 3

```
#include<iostream>
#include<cstring>
#include<stack>
using namespace std;
int main() {
    char str[20];
    char ch;
    cin >> str;
    _____; //LINE-1
    for(int i = 0; i < strlen(str); i++)
        _____; //LINE-2
    for(int i = 0; i < strlen(str) - 1; i++) {
        ch = _____; //LINE-3
        cout << ch;
        s.pop();
    }
    return 0;
}
```

Public 1

Input: program

Output: margor

Public 2

Input: welcome

Output: emocle

Private

Input: batch

Output: hcta

Answer:

```
LINE-1:  stack<char> s  
LINE-2:  s.push(str[i])  
LINE-3:  s.top()
```

Explanation:

As the stack has to work with char type. Hence at LINE-1, we need to declare it as **stack<char> s**;. At LINE-2, the elements can be pushed at stack as **s.push(str[i])**;. We need to print top of the stack for which LINE-3 will be filled as **s.top()**.

Question 2

Consider the following program.

- Fill in the blank at LINE-1 with the appropriate if statement,
- Fill in the blank at LINE-2 and LINE-3 with the appropriate return statements

The program must satisfy the sample input and output.

Marks: 3

```
#include<iostream>
using namespace std;
bool Compare(string s1, string s2){
    if(_____) //LINE-1
        _____; //LINE-2
    else
        _____; //LINE-3
}
int main(){
    string s1, s2;
    cin >> s1 >> s2;
    cout << s1 << ", " << s2 << " : " << Compare(s1, s2);
    return 0;
}
```

Public 1

Input: bat bag
Output: bat, bag : 1

Public 2

Input: C C++
Output: C, C++ : 0

Private

Input: Sir Student
Output: Sir, Student : 0

Answer:

LINE-1: s1 >= s2
LINE-2: return true
LINE-3: return false

Explanation:

At LINE-1 the condition must be `if(s1 >= s2)`, then at LINE-2 it will be `return true;`, otherwise it will be `return false;` at LINE-3.

Question 3

Consider the program below.

- Fill in the blank at LINE-1 to include appropriate header file to utilize `abs(·)` function.
- Fill in the blank at LINE-2 to compute the length between two points `p1` and `p2` as $|(p1.y - p2.y)| + |(p1.x - p2.x)|$.

The program must satisfy the given test cases.

Marks: 3

```
#include <iostream>
----- //LINE-1
using namespace std;
struct point{
    int x, y;
};

double get_len(point p1, point p2){
    return -----; //LINE-2
}

int main() {
    int x1, y1, x2, y2;
    cin >> x1 >> y1 >> x2 >> y2;
    point p1, p2;
    p1.x = x1;
    p1.y = y1;
    p2.x = x2;
    p2.y = y2;
    cout << get_len(p1, p2);
    return 0;
}
```

Public 1

Input: 2 5 3 8

Output: 4

Public 2

Input: 10 20 40 20

Output: 30

Private

Input: 20 40 60 10

Output: 70

Answer:

LINE-1: `#include <cmath>`

LINE-2: `abs(p1.x - p2.x) + abs(p1.y - p2.y)`

Explanation:

The C library `math.h` can be included in C++ program as

`#include <cmath>`

At LINE-2, the formula to compute the distance between two points can be implemented as:

`abs(p1.x - p2.x) + abs(p1.y - p2.y)`