# DSA Using C++ Bootcamp Answers

## Q1. Write a program to Swap to two numbers.

Ans.

```
#include <iostream>
using namespace std;
int main()
{
int num1, num2, temp;
cout<<"Enter 1st Number: ";
cin>>num1;
cout<<"Enter 2nd Number: ";
cin>>num2;
//displaying numbers before swapping
cout<<"Before Swapping: First Number: "<<num1<<" Second Number: "<<num2;
//swapping
temp=num1;
num1=num2;
num2=temp;
//displaying numbers after swapping
cout<<"\nAfter Swapping: First Number: "<<num1<<" Second Number: "<<num2;
return 0;
}
```

## Q2. Write a program to find the largest number among three numbers entered by the user.

Ans.

```
#include <iostream>
using namespace std;
int main() {
  int a = 5 ,b = 1 ,c = 9;
  if(a>b) {
    if(a>c)
    cout<<a<<" is largest number";
    else
    cout<<c<<" is largest number";
  }else {
    if(b>c)
    cout<<b<<" is largest number";
    else
    cout<<c<<" is largest number";
 }
  return 0;
}
```

## Q3. Write a program to check whether a year entered by a user is Leap year or not.

Ans.

```
#include<iostream>
using namespace std;
int main()
{
   int yr;
   cout<<"Enter the Year: ";
   cin>>yr;
   if((yr%4==0) && (yr%100!=0))
      cout<<"\nIt is a Leap Year";
   else if(yr%400==0)
      cout<<"\nIt is a Leap Year";
   else
      cout<<"\nIt is not a Leap Year";
   cout<<endl;
   return 0;
}
```

## Q4. Write a program to display Fibonacci Series upto nth term. (Using loops)

Ans.

```
#include <iostream>
using namespace std;
int main(){
   int N, last=1, secondLast=0, current=0, i;
   cout << "Enter number of terms in Fibonacci series\n";
   cin >> N;
   /*
    *  N term = (N-1)th therm + (N-2)th term;
    * or  current term = last term + secondLast term;
    */
   for(i = 0; i < N; i++){
      if(i < 2){
         current = i;
      } else {
         current = last + secondLast;
         secondLast = last;
         last = current;
      }
      cout << current << " ";
   }
   return 0;
}
```

Q5. Write a program to check whether a number is Prime or Not.

Ans.

```
#include <iostream>
using namespace std;
int main() {
   int i, n;
   bool isPrime = true;
   cout << "Enter a positive integer: ";
   cin >> n;
   // 0 and 1 are not prime numbers
   if (n == 0 || n == 1) {
      isPrime = false;
   }
   else {
      for (i = 2; i <= n / 2; ++i) {
         if (n % i == 0) {
            isPrime = false;
            break;
         }
      }
   }
   if (isPrime)
      cout << n << " is a prime number";
   else
      cout << n << " is not a prime number";

   return 0;
}
```

Q6. Print this pattern using loops

For n=5

```
    *
   * *
  * * *
 * * * *
* * * * *
```

Ans.

```
#include <iostream>
using namespace std;
int main()
{
 int space, rows;
   cout <<"Enter number of rows: ";
```

```
    cin >> rows;
    for(int i = 1, k = 0; i <= rows; ++i, k = 0)
    {
        for(space = 1; space <= rows-i; ++space)
        {
            cout <<"  ";
        }
        while(k != 2*i-1)
        {
            cout << "* ";
            ++k;
        }
        cout << endl;
    }
    return 0;
}
```

Q7.Write a program that takes n elements from the user and displays the second largest element of an array.

Ans.

```
#include <iostream>
using namespace std;
int main(){
  int n, num[50], largest, second;
  cout<<"Enter number of elements: ";
  cin>>n;
  for(int i=0; i<n; i++){
    cout<<"Enter Array Element"<<(i+1)<<": ";
    cin>>num[i];
  }
  if(num[0]<num[1]){
    largest = num[1];
    second = num[0];
  }
  else{
    largest = num[0];
    second = num[1];
  }
  for (int i = 2; i< n ; i ++) {
    if (num[i] > largest) {
      second = largest;
      largest = num[i];
    }
    else if (num[i] > second && num[i] != largest) {
      second = num[i];
    }
```

```
   }
   cout<<"Second Largest Element in array is: "<<second;
   return 0;
}
```

## Q8. Left rotation.

Ans.

```
#include <bits/stdc++.h>
using namespace std;
string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);
vector<int> rotateLeft(int d, vector<int> arr) {
}
int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));
    string first_multiple_input_temp;
    getline(cin, first_multiple_input_temp);
    vector<string> first_multiple_input = split(rtrim(first_multiple_input_temp));
    int n = stoi(first_multiple_input[0]);
    int d = stoi(first_multiple_input[1]);
    string arr_temp_temp;
    getline(cin, arr_temp_temp);
    vector<string> arr_temp = split(rtrim(arr_temp_temp));
    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        int arr_item = stoi(arr_temp[i]);
        arr[i] = arr_item;
    }
    vector<int> result = rotateLeft(d, arr);
    for (size_t i = 0; i < result.size(); i++) {
        fout << result[i];
        if (i != result.size() - 1) {
            fout << " ";
        }
    }
    fout << "\n";
    fout.close();
    return 0;
}

string ltrim(const string &str) {
    string s(str);
    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
    );
    return s;
}
string rtrim(const string &str) {
    string s(str);
    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
        s.end()
    );
    return s;
}
vector<string> split(const string &str) {
    vector<string> tokens;
    string::size_type start = 0;
    string::size_type end = 0;
    while ((end = str.find(" ", start)) != string::npos) {
        tokens.push_back(str.substr(start, end - start));
        start = end + 1;
    }
    tokens.push_back(str.substr(start));
    return tokens;
}
```

## Q9. Grading Students

### Ans.

```
#include <bits/stdc++.h>
using namespace std;
string ltrim(const string &);
string rtrim(const string &);
vector<int> gradingStudents(vector<int> grades) {
}
int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));
    string grades_count_temp;
    getline(cin, grades_count_temp);
    int grades_count = stoi(ltrim(rtrim(grades_count_temp)));
    vector<int> grades(grades_count);
    for (int i = 0; i < grades_count; i++) {
        string grades_item_temp;
        getline(cin, grades_item_temp);
int grades_item = stoi(ltrim(rtrim(grades_item_temp)));
        grades[i] = grades_item;
    }
    vector<int> result = gradingStudents(grades);
    for (size_t i = 0; i < result.size(); i++) {
        fout << result[i];
        if (i != result.size() - 1) {
            fout << "\n";
        }
    }
    fout << "\n";
    fout.close();
    return 0;
}
string ltrim(const string &str) {
    string s(str);
    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
    );
    return s;
}
string rtrim(const string &str) {
    string s(str);
    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
        s.end()
    );
    return s;
}
```

## Q10. Camel Case

Ans.

```
#include <bits/stdc++.h>
using namespace std;
int camelcase(string s) {
}
int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));
    string s;
    getline(cin, s);
    int result = camelcase(s);
    fout << result << "\n";
    fout.close();
 return 0;
}
```