**DSA BOOTCAMP ASSIGNMENT**

Q1. Write a program to Swap to two numbers.

Code:

#include<stdio.h>

#include<stdlib.h>

int main()

{

int a=10, b=20;

printf("Before swap a=%d b=%d",a,b);

a=a\*b;//a=200 (10\*20)

b=a/b;//b=10 (200/20)

a=a/b;//a=20 (200/10)

system("cls");

printf("\nAfter swap a=%d b=%d",a,b);

return 0;

}

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

Code:

#include <stdio.h>

int main()

{

int num1, num2, num3;

printf(" Enter the number1 = ");

scanf("%d", &num1);

printf("\n Enter the number2 = ");

scanf("%d", &num2);

printf("\n Enter the number3 = ");

scanf("%d", &num3);

if (num1 > num2)

{

if (num1 > num3)

{

printf("\n Largest number = %d \n",num1);

}

else

{

printf("\n Largest number = %d \n",num3);

}

}

else if (num2 > num3)

{

printf("\n Largest number = %d \n",num2);

}

else

{

printf("\n Largest number = %d \n",num3);

}

return 0;

}

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

Code:

#include <stdio.h>

int main() {

int year;

printf("Enter a year: ");

scanf("%d", &year);

// leap year if perfectly divisible by 400

if (year % 400 == 0) {

printf("%d is a leap year.", year);

}

// not a leap year if divisible by 100

// but not divisible by 400

else if (year % 100 == 0) {

printf("%d is not a leap year.", year);

}

// leap year if not divisible by 100

// but divisible by 4

else if (year % 4 == 0) {

printf("%d is a leap year.", year);

}

// all other years are not leap years

else {

printf("%d is not a leap year.", year);

}

return 0;

}

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

Code:

#include<stdio.h>

int main()

{

int sum = 0, n;

int a = 0;

int b = 1;

printf("Enter the nth value: ");

scanf("%d", &n);

printf("Fibonacci series: ");

while(sum <= n)

{

printf("%d ", sum);

a = b; // swap elements

b = sum;

sum = a + b; // next term is the sum of the last two terms

}

return 0;

}

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

Code:

#include <stdio.h>

int main() {

int n, i, flag = 0;

printf("Enter a positive integer: ");

scanf("%d", &n);

for (i = 2; i <= n / 2; ++i) {

// condition for non-prime

if (n % i == 0) {

flag = 1;

break;

}

}

if (n == 1) {

printf("1 is neither prime nor composite.");

}

else {

if (flag == 0)

printf("%d is a prime number.", n);

else

printf("%d is not a prime number.", n);

}

return 0;

}

Q6. Print this pattern using loops

For n=5

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Code:

#include <stdio.h>

int main() {

int i, space, rows, k = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i, k = 0) {

for (space = 1; space <= rows - i; ++space) {

printf(" ");

}

while (k != 2 \* i - 1) {

printf("\* ");

++k;

}

printf("\n");

}

return 0;

}

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

Code:

#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. <https://www.hackerrank.com/challenges/array-left-rotation/problem>

A left rotation operation on an array of size **n** shifts each of the array's elements **1** unit to the left. Given an integer, **d** , rotate the array that many steps left and return the result.

Code:

#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. <https://www.hackerrank.com/challenges/grading/problem>

Code:

#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. <https://www.hackerrank.com/challenges/camelcase/problem>

Code:

#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;

}