**DSA BOOTCAMP ASSIGNMENT**

Q1. Write a program to Swap to two numbers.

#include<stdio.h>

**int** main()

{

**int** a=10, b=20;

printf("Before swap a=%d b=%d",a,b);

a=a+b;//a=30 (10+20)

b=a-b;//b=10 (30-20)

a=a-b;//a=20 (30-10)

printf("\nAfter swap a=%d b=%d",a,b);

**return** 0;

}

**Output:**

Before swap a=10 b=20

After swap a=20 b=10

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

#include <stdio.h>

int main() {

double n1, n2, n3;

printf("Enter three different numbers: ");

scanf("%lf %lf %lf", &n1, &n2, &n3);

// if n1 is greater than both n2 and n3, n1 is the largest

if (n1 >= n2 && n1 >= n3)

printf("%.2f is the largest number.", n1);

// if n2 is greater than both n1 and n3, n2 is the largest

if (n2 >= n1 && n2 >= n3)

printf("%.2f is the largest number.", n2);

// if n3 is greater than both n1 and n2, n3 is the largest

if (n3 >= n1 && n3 >= n2)

printf("%.2f is the largest number.", n3);

return 0;

}

**OUTPUT**

Enter three numbers: -4.5

3.9

5.6

5.60 is the largest number.

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

#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;

}

**Output**

Enter a year: 1900

1900 is not a leap year.

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

#include <iostream>

using namespace std;

int main() {

int n, t1 = 0, t2 = 1, nextTerm = 0;

cout << "Enter the number of terms: ";

cin >> n;

cout << "Fibonacci Series: ";

for (int i = 1; i <= n; ++i) {

// Prints the first two terms.

if(i == 1) {

cout << t1 << ", ";

continue;

}

if(i == 2) {

cout << t2 << ", ";

continue;

}

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;

cout << nextTerm << ", ";

}

return 0;

}

**Output**

Enter the number of terms: 10

Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

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

#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;

}

**Output**

Enter a positive integer: 29

29 is a prime number.

Q6. Print this pattern using loops

For n=5

    \*

  \* \*

  \* \* \*

\* \* \* \*

\* \* \* \* \*

#include<stdio.h>

int main()

{

int n,r,c;

char ch = 'A';

printf("Enter number of rows: ");

scanf("%d",&n);

// outer loop

for(r=1; r<=n; r++)

{

// inner loop

for(c=1; c<=n; c++)

{

if(c==1||r==1||c==n||r==n) printf("\* ");

else printf("%c ",ch++);

if(ch > 'Z') ch='A';

} // end of inner loop

printf("\n");

} // end of outer loop

return 0;

}

**OUTPUT**

    \*

  \* \*

  \* \* \*

\* \* \* \*

\* \* \* \* \*

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

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;    /\* Function to print the second largest elements \*/  void print2largest(int arr[], int arr\_size)  {      int i, first, second;        /\* There should be atleast two elements \*/      if (arr\_size < 2) {          printf(" Invalid Input ");          return;      }        // sort the array      sort(arr, arr + arr\_size);        // start from second last element      // as the largest element is at last      for (i = arr\_size - 2; i >= 0; i--) {          // if the element is not          // equal to largest element          if (arr[i] != arr[arr\_size - 1]) {              printf("The second largest element is %d\n", arr[i]);              return;          }      }        printf("There is no second largest element\n");  }    /\* Driver program to test above function \*/  int main()  {      int arr[] = { 12, 35, 1, 10, 34, 1 };      int n = sizeof(arr) / sizeof(arr[0]);      print2largest(arr, n);      return 0;  } |

**Output:**

The second largest element is 34