All Practical Questions – Aryan Sethi

Q1)

#include <iostream>

using namespace std;

int main()

{

int tmp, last\_digit, product, sum;

product = 1;

sum = 0;

cout << "Enter a number: " << endl;

cin >> tmp;

int n = tmp < 0 ? (-tmp) : tmp;

while (n > 0)

{

last\_digit = n % 10;

product \*= last\_digit;

sum += last\_digit;

n /= 10;

}

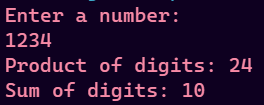
cout << "Product of digits: " << product << endl;

cout << "Sum of digits: " << sum << endl;

return 0;

}

OUTPUT



Q2)

#include <iostream>

using namespace std;

int main()

{

int tmp, n, reversed, last\_digit;

reversed = 0;

cout << "Enter the number: ";

cin >> tmp;

n = tmp < 0 ? -tmp : tmp;

while (n > 0)

{

last\_digit = n % 10;

reversed = 10 \* reversed + last\_digit;

n /= 10;

}

reversed = tmp < 0 ? -reversed : reversed;

cout << "Reversed number: " << reversed;

return 0;

}

OUTPUT



Q3)

#include <iostream>

using namespace std;

int main()

{

int n;

cout << "Enter value of n: ";

cin >> n;

if (n <= 0){

cout << "n can not be less than or equal to zero";

exit(1);

}

float sum = 1;

for (int i = 2; i <= n; i++)

{

sum += (1.0/i);

}

cout << "Sum of series: " << sum;

return 0;

}

OUTPUT





Q4)

#include <iostream>

using namespace std;

int main()

{

int n;

cout << "Enter value of n: ";

cin >> n;

if (n <= 0){

cout << "n can not be less than or equal to zero";

exit(1);

}

int sum = 0;

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

{

if (i % 2 == 0){

sum -= i;

}

else{

sum += i;

}

}

cout << "Sum of series: " << sum;

return 0;

}

OUTPUT





Q5)

#include <iostream>

using namespace std;

bool isPalindrome(string s){

int n = s.size();

for (int i = 0; i < n/2; i++)

{

if (s[i] != s[n - i - 1])

return false;

}

return true;

}

int main()

{

string s;

cin >> s;

if (isPalindrome(s))

cout << s << " is a palindrome" << endl;

else

cout << s << " is not a palindrome" << endl;

return 0;

}

OUTPUT



Q6)

#include <iostream>

using namespace std;

bool is\_prime(int n){

for (int i = 2; i < n; i++)

{

if (n % i == 0){

return false;

}

}

return true;

}

int main()

{

for (int i = 2; i < 100; i++)

{

if (is\_prime(i)){

cout << i << endl;

}

}

return 0;

}

OUTPUT



Q7)

#include <iostream>

using namespace std;

int main()

{

int n;

cout << "Enter a number: ";

cin >> n;

if (n <= 0){

cout << "n can not be negative or zero" ;

exit(1);

}

cout << "Factors of " << n << endl;

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

{

if (n % i == 0){

cout << i << " ";

}

}

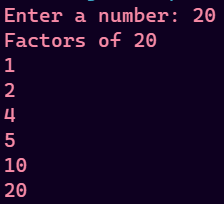
cout << endl;

return 0;

}

OUTPUT

****

****

Q9)

#include <iostream>

using namespace std;

int main()

{

int n;

cin >> n;

if (n <= 0){

cout << "n can not be negative or less than zero";

exit(1);

}

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

{

for (int j = 1; j <= (2\*i - 1); j++)

{

cout << "\* ";

}

cout << endl;

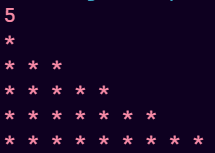
}

return 0;

}

output

****

****

#include <iostream>

using namespace std;

void printArray(int arr[], int n)

{

cout << "Array -> ";

for (int i = 0; i < n; i++)

{

cout << arr[i] << " ";

}

cout << endl;

}

void print\_even(int arr[], int n)

{

for (int i = 0; i < n; i++)

{

if (arr[i] % 2 == 0)

{

cout << arr[i] << " ";

}

}

cout << endl;

}

void print\_odd(int arr[], int n)

{

for (int i = 0; i < n; i++)

{

if (arr[i] % 2 != 0)

{

cout << arr[i] << " ";

}

}

cout << endl;

}

void statistics(int arr[], int n)

{

int sum = 0;

for (int i = 0; i < n; i++)

{

sum += arr[i];

}

float average = float(sum) / n;

cout << "Average: " << average << endl;

cout << "Sum: " << sum << endl;

}

void max\_min(int arr[], int n)

{

int max, min;

max = arr[0];

min = arr[0];

for (int i = 0; i < n; i++)

{

if (arr[i] > max)

{

max = arr[i];

}

if (arr[i] < min)

{

min = arr[i];

}

}

cout << "Max element: " << max << endl;

cout << "Min element: " << min << endl;

}

int remove\_duplicates(int arr[], int n)

{

for (int i = 0; i < n; i++)

{

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

{

if (arr[i] == arr[j])

{

for (int k = j; k < n - 1; k++)

{

arr[k] = arr[k + 1];

}

n--;

}

}

}

return n;

}

void print\_reverse(int arr[], int n)

{

cout << "Array in reverse -> ";

for (int i = n - 1; i >= 0; i--)

{

cout << arr[i] << " ";

}

cout << endl;

}

int main()

{

int n;

char choice;

cout << "Enter size of array: ";

cin >> n;

if (n <= 0)

{

cout << "Array should have atleast one element" << endl;

exit(1);

}

int arr[n];

cout << "Enter elements of array" << endl;

for (int i = 0; i < n; i++)

{

cin >> arr[i];

}

while (true)

{

cout << "------MENU------" << endl;

cout << "1. Print even valued elements" << endl;

cout << "2. Print odd valued elements" << endl;

cout << "3. Average and Sum of elements" << endl;

cout << "4. Maximum and Minimum among the elements" << endl;

cout << "5. Remove duplicates" << endl;

cout << "6. Array in Reverse Order" << endl;

cout << "7. Print Array" << endl;

cout << "8. EXIT" << endl;

cout << "Make your choice: ";

cin >> choice;

if (choice == '8')

{

cout << "Exiting Program" << endl;

break;

}

else if (choice == '1')

{

print\_even(arr, n);

}

else if (choice == '2')

{

print\_odd(arr, n);

}

else if (choice == '3')

{

statistics(arr, n);

}

else if (choice == '4')

{

max\_min(arr, n);

}

else if (choice == '5')

{

cout << "Before duplicate removal" << endl;

printArray(arr, n);

n = remove\_duplicates(arr, n);

cout << "After duplicate removal" << endl;

printArray(arr, n);

}

else if (choice == '6')

{

print\_reverse(arr, n);

}

else if (choice == '7')

{

printArray(arr, n);

}

else

{

cout << "Invalid choice" << endl;

}

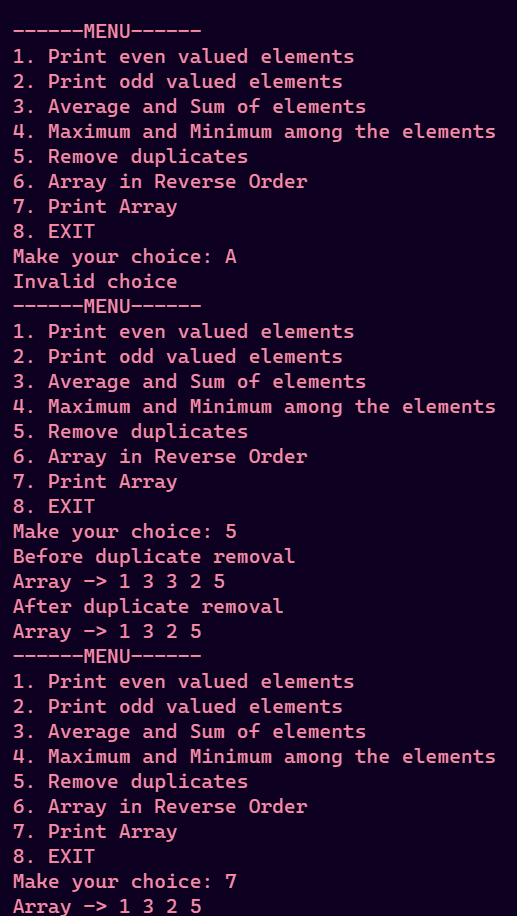
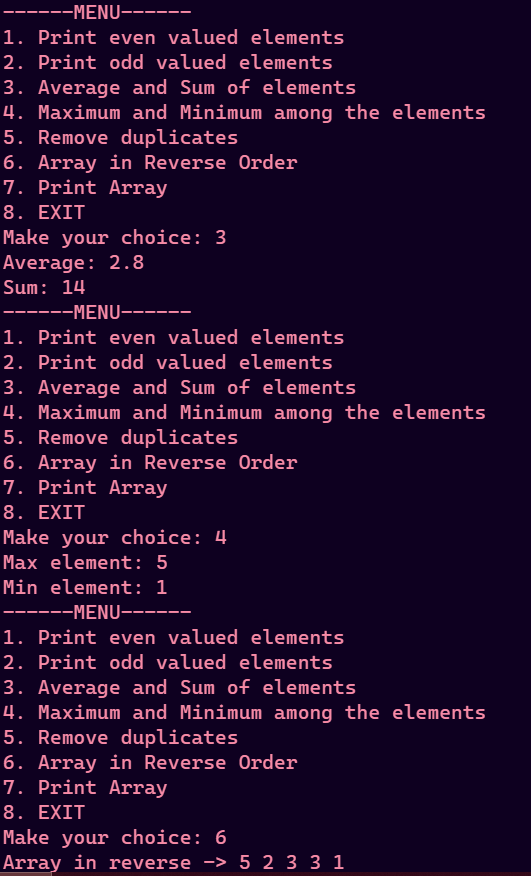
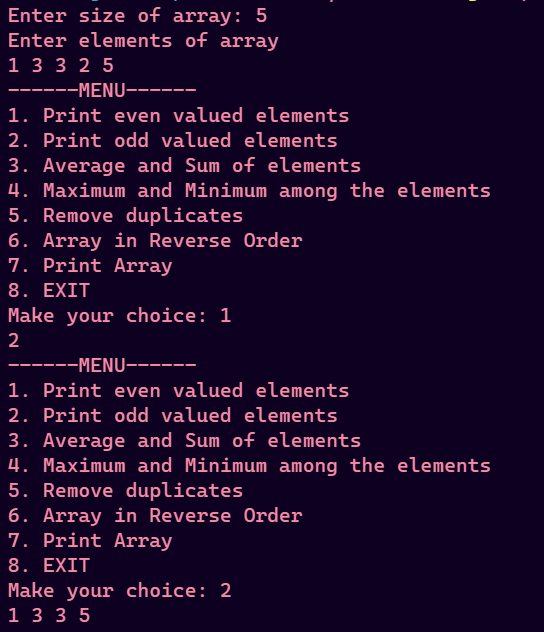
}

return 0;

}

output

****

****

Q11)

#include <iostream>

using namespace std;

int main(int argCount, char \*\*args)

{

if (argCount <= 1)

{

cout << "No command line arguments entered" << endl;

exit(1);

}

string sentence = args[1];

// A - Z : 65 to 90

// a - z : 97 to 122

int charCount[123];

for (int i = 0; i < 123; i++)

charCount[i] = 0;

for (int i = 0; i < sentence.size(); i++)

charCount[sentence[i]]++;

// Printing table

cout << "Alphabet : Count" << endl;

for (int i = 0; i < 123; i++)

{

if (charCount[i])

{

char c = i;

cout << c << " : " << charCount[i] << endl;

}

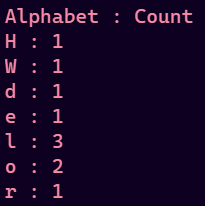
}

return 0;

}

output





Q12)

#include <iostream>

using namespace std;

void swapNums(int \*a , int \*b){

int tmp = \*a;

\*a = \*b;

\*b = tmp;

}

int main()

{

int a , b;

cin >> a >> b;

cout << "Initially" << endl;

cout << "a : " << a << " , b : " << b << endl;

swapNums(&a , &b);

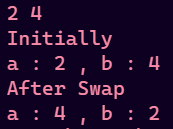
cout << "After Swap" << endl;

cout << "a : " << a << " , b : " << b << endl;

return 0;

}

output



Q13)

#include <iostream>

using namespace std;

void alter(int \*a, int \*b)

{

int tmp = \*b;

\*b = \*a + \*b;

\*a = tmp;

}

int main()

{

int a, b;

cin >> a >> b;

cout << "Initially" << endl;

cout << "a : " << a << " , b : " << b << endl;

alter(&a, &b);

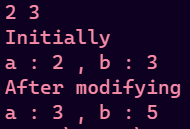
cout << "After modifying" << endl;

cout << "a : " << a << " , b : " << b << endl;

return 0;

}

output



Q14)

#include <iostream>

using namespace std;

float area(float r){

return 3.14 \* r \* r;

}

float circumference(float r){

return 3.14 \* r \* 2;

}

int main()

{

float r;

cout << "Enter radius of circle: ";

cin >> r;

if (r < 0){

cout << "Radius can not be negative" << endl;

exit(1);

}

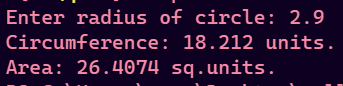
cout << "Circumference: " << circumference(r) << " units." << endl;

cout << "Area: " << area(r) << " sq.units." << endl;

return 0;

}

Output

****

Q16)

#include <iostream>

#include <cstring>

using namespace std;

void showAddress(string s)

{

void \*p;

for (int i = 0; i < s.size(); i++)

{

p = &s[i];

cout << s[i] << " : " << p << endl;

}

}

string concatStr(string s1, string s2)

{

return s1 + s2;

}

int lengthOfStr(string s)

{

int length = 0;

char \*p = &s[0];

while (\*p != '\0')

{

length++;

p++;

}

return length;

}

void toUpperCase(string &s)

{

for (int i = 0; i < s.size(); i++)

{

if (s[i] >= 'a' && s[i] <= 'z')

s[i] -= 32;

}

}

void toLowerCase(string &s)

{

for (int i = 0; i < s.size(); i++)

{

if (s[i] >= 'A' && s[i] <= 'Z')

s[i] += 32;

}

}

void countVowels(string s)

{

int count[] = {0, 0, 0, 0, 0};

for (int i = 0; i < s.size(); i++)

{

switch (s[i])

{

case 'a':

count[0]++;

break;

case 'A':

count[0]++;

break;

case 'e':

count[1]++;

break;

case 'E':

count[1]++;

break;

case 'i':

count[2]++;

break;

case 'I':

count[2]++;

break;

case 'o':

count[3]++;

break;

case 'O':

count[3]++;

break;

case 'u':

count[4]++;

break;

case 'U':

count[4]++;

break;

default:

break;

}

}

cout << "Vowel occurences in " << s << endl;

cout << "a : " << count[0] << endl;

cout << "e : " << count[1] << endl;

cout << "i : " << count[2] << endl;

cout << "o : " << count[3] << endl;

cout << "u : " << count[4] << endl;

}

void reverseString(string &s)

{

int n = s.size();

char tmp;

for (int i = 0; i < n / 2; i++)

{

tmp = s[i];

s[i] = s[n - i - 1];

s[n - i - 1] = tmp;

}

}

int main()

{

// string s1 = "Hello";

// string s2 = " World";

// char s3[s1.size()+1];

// for (int i = 0; i < s1.size(); i++)

// s3[i] = s1[i];

// s3[s1.size()] = '\0';

// char s3[200] = "Aryan";

// char s4[200] = "Aryan";

// cout << concatStr2(s3 , s4);

// cout << strcmp(s3 , s4);

// cout << lengthOfStr(s1) << endl;

// cout << lengthOfStr(s3) << endl;

// toUpperCase(s1);

// reverseString(s1);

// cout << s1;

// countVowels(s3);

int choice;

string s1, s2;

char s3[200], s4[200];

int cmp;

while (true)

{

cout << "-----------MENU-----------" << endl;

cout << "1.Show address of each character in string" << endl;

cout << "2.Concatenate two strings (w/o strcat)" << endl;

cout << "3.Concatenate two strings (w/ strcat)" << endl;

cout << "4.Compare two strings" << endl;

cout << "5.Length of string" << endl;

cout << "6.Convert all lowercase characters to uppercase" << endl;

cout << "7.Convert all uppercase characters to lowercase" << endl;

cout << "8.Number of vowels" << endl;

cout << "9.Reverse a string" << endl;

cout << "10.EXIT" << endl;

cout << "Your choice : ";

cin >> choice;

if (choice == 10)

{

cout << "Exiting program" << endl;

break;

}

switch (choice)

{

case 1:

cout << "Enter a string: ";

cin >> s1;

cout << "Your string: " << s1 << endl;

showAddress(s1);

break;

case 2:

cout << "Enter two strings: ";

cin >> s1 >> s2;

cout << "String 1 : " << s1 << endl;

cout << "String 2 : " << s2 << endl;

cout << "Concatenated String : " << concatStr(s1, s2) << endl;

break;

case 3:

cout << "Enter two strings: ";

cin >> s3 >> s4;

cout << "String 1 : " << s3 << endl;

cout << "String 2 : " << s4 << endl;

cout << "Concatenated String : " << strcat(s3, s4) << endl;

break;

case 4:

cout << "Enter two strings: ";

cin >> s3 >> s4;

cout << "String 1 : " << s3 << endl;

cout << "String 2 : " << s4 << endl;

cmp = strcmp(s3, s4);

if (!cmp)

cout << "The two strings are equal" << endl;

else if (cmp > 0)

cout << s3 << " is lexographically greater than " << s4 << endl;

else

cout << s4 << " is lexographically greater than " << s3 << endl;

break;

case 5:

cout << "Enter a string: ";

cin >> s1;

cout << "Length of " << s1 << " : " << lengthOfStr(s1) << endl;

break;

case 6:

cout << "Enter a string: ";

cin >> s1;

cout << "Orignal string: " << s1 << endl;

toUpperCase(s1);

cout << "Uppercase string: " << s1 << endl;

break;

case 7:

cout << "Enter a string: ";

cin >> s1;

cout << "Orignal string: " << s1 << endl;

toLowerCase(s1);

cout << "Lowercase string: " << s1 << endl;

break;

case 8:

cout << "Enter a string: ";

cin >> s1;

countVowels(s1);

break;

case 9:

cout << "Enter a string: ";

cin >> s1;

cout << "Orignal string: " << s1 << endl;

reverseString(s1);

cout << "Reversed string: " << s1 << endl;

break;

default:

cout << "Invalid choice" << endl;

break;

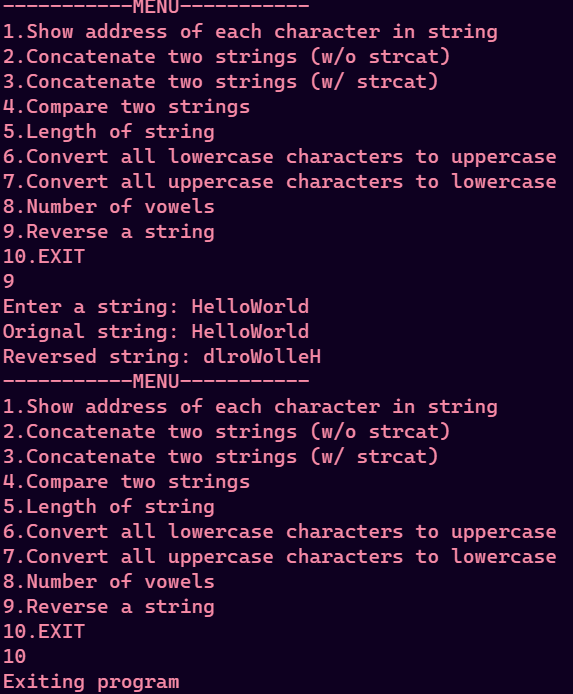
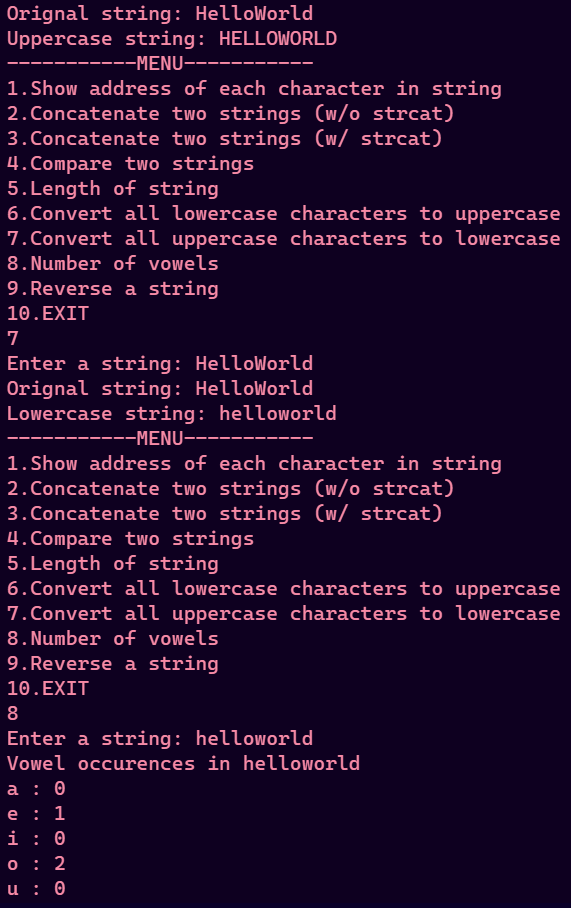
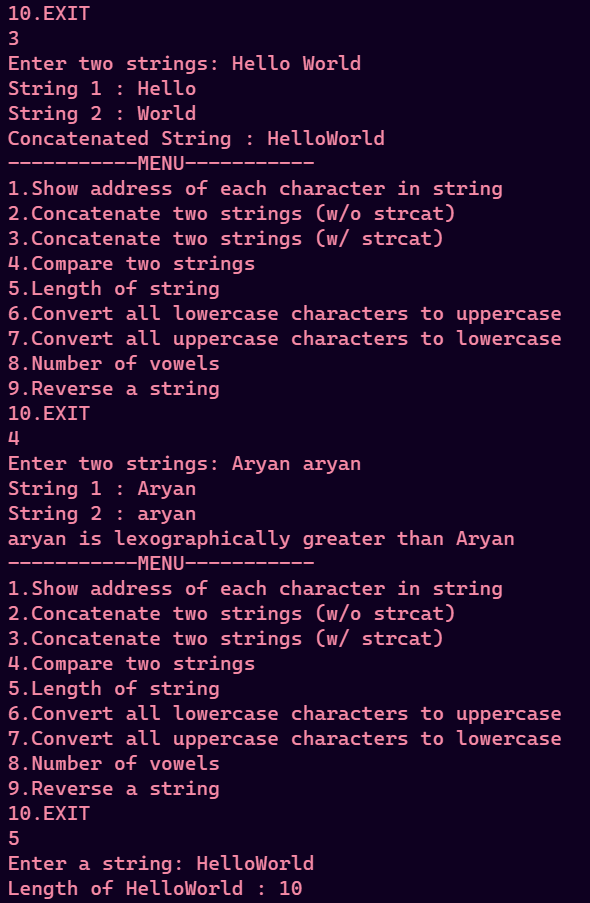
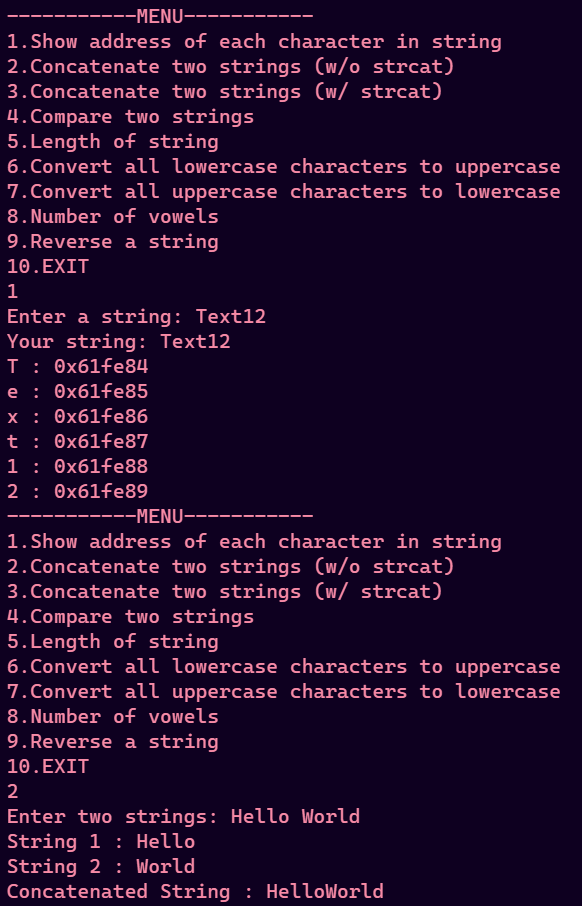
}

}

return 0;

}

output



Q17)

#include <iostream>

using namespace std;

void print\_arr(int arr[] , int n){

for (int i = 0; i < n; i++)

{

cout << arr[i] << " ";

}

cout << endl;

}

int main()

{

int n , m , p;

cout << "Enter lengths of array 1 and 2: " << endl;

cin >> n >> m;

if (n <= 0 || m <= 0){

cout << "Array length can not be zero or negative" << endl;

exit(1);

}

int arr1[n];

int arr2[m];

p = n + m;

int arr[p];

cout << "Enter elements of array 1" << endl;

for (int i = 0; i < n; i++)

{

cin >> arr1[i];

}

cout << "Enter elements of array 2" << endl;

for (int i = 0; i < m; i++)

{

cin >> arr2[i];

}

// Merging

int i = 0 , j = 0 , k = 0;

while (i < n || j < m){

if (i < n && j < m ){

if (arr1[i] < arr2[j]){

arr[k] = arr1[i];

i++;

}

else{

arr[k] = arr2[j];

j++;

}

}

else if (i < n){

arr[k] = arr1[i];

i++;

}

else{

arr[k] = arr2[j];

j++;

}

k++;

}

cout << "Array 1: ";

print\_arr(arr1 , n);

cout << "Array 2: ";

print\_arr(arr2 , m);

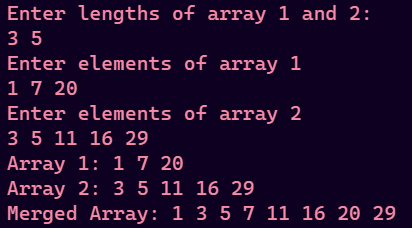
cout << "Merged Array: ";

print\_arr(arr , p);

return 0;

}

output

****

Q18)