1). Write a function to reverse a singly linked list.The function should take the head of the list and return the new head of the reversed list.

Code:

#include<bits/stdc++.h>

using namespace std;

class Node

{

public:

int data;

Node \* next;

Node(int x){

data = x;

next = NULL;

}

};

Node\* head = NULL;

void insert(int x)

{

Node\* new\_node = new Node(x);

if(head == NULL){

head = new\_node;

return;

}

new\_node->next = head;

head = new\_node;

return;

}

void reverse()

{

Node\* current = head;

Node\* prev = NULL;

Node\* next;

while(current != NULL)

{

next = current->next;

current->next = prev;

prev = current;

current = next;

}

head = prev;

}

void print()

{

Node\* temp = head;

while(temp != NULL)

{

cout<<temp->data;

temp = temp->next;

}

return;

}

int main()

{

int n , x;

cout<< "Enter the size of linked list: ";

cin>>n;

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

{

cout<<"Enter the element at " << i << " : ";

cin>> x;

insert(x);

}

cout<<"Before reversing: " ;

print();

cout<<"\n";

reverse();

cout<<"After reversing: " ;

print();

}

output:



2) Given a string, find the length of the longest substring without repeating characters.The function should return an integer representing the length of the longest substring without repeating characters.

Code:

#include<bits/stdc++.h>

using namespace std;

int longest\_substr(string s)

{

unordered\_map<char , int> mp;

int i = 0;

int maxlen = -1;

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

{

mp[s[j]]++;

while(mp[s[j]] > 1){

mp[s[i++]]--;

}

maxlen = max(maxlen , j - i + 1);

}

return maxlen;

}

int main()

{

string s = "";

cout<< "Enter the string: ";

cin>>s;

int result = longest\_substr(s);

cout<<"Size of the longest substring without unique characters " << result;

}

Output:



5)Write a function to rotate an array to the right by k steps.The function should modify the array in place to achieve the rotation.

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n , k;

cout<< "Enter the size of the array: ";

cin>> n;

cout<< "How many times to rotate: ";

cin>>k;

int arr[n];

k = k % n;

if(k < 0)

{

cout<< "Enter a valid k times to rotate";

return 0;

}

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

{

cout<<"Enter the " << i << " element";

cin>>arr[i];

}

reverse(arr , arr+k);

reverse(arr+k , arr+n);

reverse(arr , arr+n);

cout<<"After rotating the array by k steps"<<endl;

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

{

cout<<arr[i];

}

}

Output:



6)Write a function to find the factorial of a given number.The function should return the factorial of the number.

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cout<<"Enter the number: ";

cin>>n;

if(n == 0)

{

cout<<1;

return 0;

}

if(n < 0)

{

cout<< "not defined";

return 0;

}

int fact = 1;

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

{

fact = fact \* i;

}

cout<< "Factorial of the number is : "<<fact;

}

Output:



7) Write a function to compute the sum of the digits of a given number.The function should return the sum of the digits of the number.

Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int num;

cout<< "Enter any number";

cin>>num;

int sum = 0;

while(num > 0)

{

sum += num % 10;

num /= 10;

}

cout<< "Sum of the digits of numbers is" << sum;

}

Output:



8) Write a function to find the greatest common divisor (GCD) of two numbers. The function should return the GCD of a and b.

Code:

#include<bits/stdc++.h>

using namespace std;

int gcd(int num1 , int num2)

{

while(num2 != 0)

{

int rem = num1 % num2;

num1 = num2;

num2 = rem;

}

return num1;

}

int main()

{

int num1 , num2;

cout<< "Enter the 2 numbers : ";

cin>> num1 >> num2;

int res = gcd(num1, num2);

cout<< "Gcd of 2 numbers" << res;

}

Output:



9) Write a function to find the maximum difference between any two elements in an array.The function should return the maximum difference between any two elements in the array.

Code:

#include<bits/stdc++.h>

using namespace std;

int max\_diff(int arr[] , int n)

{

int minimum = INT\_MAX;

int maximum = INT\_MIN;

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

{

if(arr[i] > maximum){

maximum = arr[i];

}

if(arr[i] < minimum){

minimum = arr[i];

}

}

return maximum - minimum;

}

int main()

{

int n;

cout<<"Enter the size of the array:";

cin>>n;

if(n == 1){

cout<<"Size of the array has to be greater than 1";

}

int arr[n];

cout<<"Enter the array elements";

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

{

cin>>arr[i];

}

cout<< "maximum difference of two elements: " <<max\_diff(arr , n);

}

Output:



10) Write a function to check if a given string contains only alphabetic characters.The function should return true if the string contains only alphabetic characters, and false otherwise.

Code:

#include<bits/stdc++.h>

using namespace std;

bool check\_alphabets(string s)

{

bool flag = true;

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

{

int ascii\_value = s[i] - 0;

if(ascii\_value > 90 && ascii\_value < 97 || ascii\_value < 65 || ascii\_value > 122 )

{

return false;

}

}

return true;

}

int main()

{

string s = "";

cout<< "Enter characters: ";

cin>>s;

cout<< check\_alphabets(s);

}

Output:

