CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Computer Science & Engineering

Course No: CSE- 244

Course Title:
Algorithm Design & Analysis (Sessional)

Experiment No: 03

Name Of the Experiment: Divide and Conquer.

Identity Details

Name of the Student: Md. Safayat Bin Nasir

Student Id Number: 2004099

Level: 2 Term: 2 Section: B

Group: B1

Date of performance: 24-09-2023

Date of Submission: 01-10-2023

Remarks

Program 1: Given two integers x and n, efficiently calculate the value of x^n.

```cpp
#include <bits/stdc++.h>
using namespace std;
#define sad '\n'
#define all(b) b.begin(), b.end()
long long t = 1;
int power(int x, int n) {
  if (n == 1) return x ;
  int y = power(x, n / 2);
  if (n % 2 == 0) {
    return 1LL * y * y ;
  }
  else {
    return (1LL * y * y)  * x ;
  }
}
int main()
{
    int  x, n ;
    while(cin >> x >> n){
        cout << power(x , n) << '\n';
    }
}
```

Output:

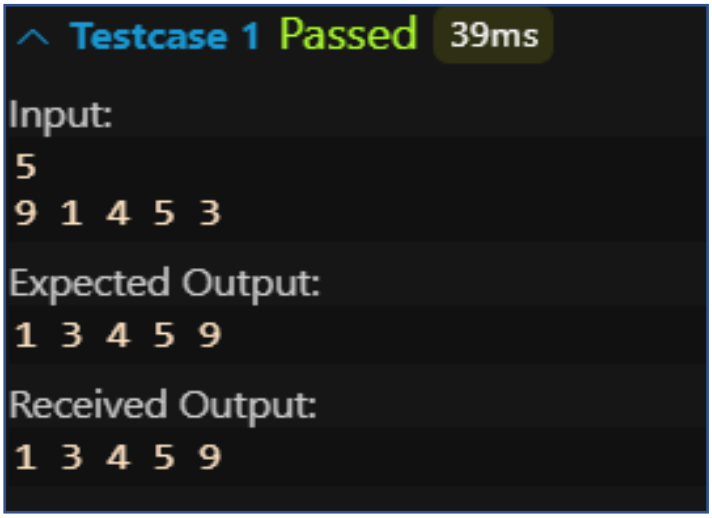
Program 2: Sort an array using merge sort algorithm.

```cpp
#include <bits/stdc++.h>
using namespace std;
#define sad '\n'
void merge(vector<int>&v , int  l , int mid ,int r){
    //cout << l << " "<< mid <<  " " << r << sad;
    vector<int>left, right;
    for(int i = 0; i < mid - l + 1; i++){
        left.push_back(v[l+i]);
    }
    for(int i = 0; i < r - mid ; i++){
        right.push_back(v[mid + 1 + i]);
    }
    int k = l , i , j;
    for(i = 0 , j  = 0 ; i < mid - l + 1 and j < r - mid; k++){
        if(left[i] <= right[j]){
            v[k] = left[i];
            i++;
        }
        else{
            v[k] = right[j];
            j++;
        }
    }
    //cout << i << " " << j << sad;
```

Program 2: Sort an array using merge sort algorithm.

```
    while(i < (mid - l + 1) ){
        v[k] = left[i];
        i++;
        k++;
    }
    while(j < r - mid  ){
        v[k] = right[j];
        k++ ; j++;
    }
}
void mergesort(vector<int>&v , int l , int r){
    if(l == r) return;
    //cout <<mid << sad;
    int mid = l + (r - l) / 2;
    mergesort(v , l , mid);
    mergesort(v , mid+ 1 , r);
    merge(v , l , mid , r);
}
int main()
{
    int n ;
    cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    mergesort(v , 0 , n);
    for(auto &x:v) cout << x << " ";
}
```

Output:



Program 3: Given an array of integers, find maximum sum subarray among all

```cpp
#include <bits/stdc++.h>
using namespace std;
int maxsum(vector<int>&v , int l , int mid , int r ){
    int sum = 0 , lsum = INT_MIN, rsum = INT_MIN;
    for(int i = mid; i>= l ; i--){
        sum += v[i];
        if(sum > lsum) lsum = sum;
    }
    sum = 0 ;
    for(int i = mid; i<= r; i++){
        sum += v[i];
        if(sum > rsum) rsum = sum;
    }
    return max({lsum , rsum , lsum + rsum - v[mid]});
}
int subarraysum(vector<int>&v , int l , int r){
    if(l > r) return INT_MIN;
    if(l == r) return v[l];
    int mid = l + ( r - l) / 2;
    return max({subarraysum(v , l , mid-1) , subarraysum(v , mid+1 , r) , maxsum(v , l , mid , r)});
}
int main()
{
    int n ; cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    cout << subarraysum(v , 0 , n-1 )<< endl;

}
```

Output: