

Digital assignment-1

DAA



February 2, 2024

Siddharth sanjeev

22BCE2930

Problem 1:

Develop a program to read an array of n elements and print the maximum subarray sum

I/P Format:-

Read the number of elements (n) in the array in line 1

Read n numbers in the next n lines

O/P Format:-

Print the maximum subarray sum

Code:

#include <bits/stdc++.h>

using namespace std;

int CSS(int negInf,int vec[], int low, int high, int mid){

int s = 0;

int left = negInf;

for(int i = mid;i>=low;i--){

s += vec[i];

if(s>left) left = s;

}

s = 0;

int right = negInf;

for(int i = mid;i<=high;i++){

s += vec[i];

if(s > right) right = s;

}

return max(left + right-vec[mid],max(right,left));

}

int MSA(int negInf,int vec[],int low, int high){

if(low==high) return vec[low];

else{

int mid = (low+high)/2;

int left = MSA(negInf,vec,low,mid);

int right = MSA(negInf,vec,mid+1,high);

int cross = CSS(negInf,vec,low,high,mid);

return max(left,max(right,cross));

}

}

int main(){

int size;

int negInf=0;

cin>>size;

int vec[size];

int high = size-1;

int i=0;

while(size--){

int x;

cin>>x;

vec[i++] = x;

if(negInf > x) negInf = x;

}

--negInf;

cout << MSA(negInf,vec,0,high) << endl;

return 0;

}

Problem 2:

Implement the Karatsuba algorithm for faster integer multiplication.

Input:- Two integers

Output:- Their product using Karatsuba Algorithm

Code:

#include <bits/stdc++.h>

using namespace std;

long findLength(long a, long b)

{

long count = 0;

while (a > 0 && b > 0)

{

count++;

a /= 10;

b /= 10;

}

return count;

}

long karatsuba(long a, long b, long len)

{

if (len <= 1)

{

return a \* b;

}

long x = len / 2;

long y = (long)pow(10, x);

long a0 = a % y;

long a1 = a / y;

long b0 = b % y;

long b1 = b / y;

long z2 = karatsuba(a1, b1, x);

long z0 = karatsuba(a0, b0, x);

long z1 = karatsuba(a1 + a0, b0 + b1, x) - z2 - z0;

return z2 \* y \* y + z1 \* y + z0;

}

int main()

{

long a, b;

cin >> a >> b;

long len = findLength(a, b);

cout << karatsuba(a, b, len) << endl;

return 1;

}

Problem 3:

A farmer used to cultivate vegetables in his field and sell them in a market 50 km away. The farmer has some containers of certain fixed capacities. Out of these, exactly one will be available every day and the remaining will be under circulation in the market. Depending on the capacity of the container available on a particular day, the farmer has to maximize his daily profit. Write a program to do this with the following input and output requirements.

Input Format:

Read the number of vegetable types (n) available on a particular day in line 1 [1,2,..,n]

Read the weight of each vegetable type in the next n lines

Read the profit that he gets by selling each vegetable type in the subsequent n lines

Read the capacity of the container available on that day

Output Format:

Print the total profit (Rounded off to 2 decimal places) that the farmer gets on that particular day

Code:

#include <algorithm>

#include <vector>

#include <iostream>

using namespace std;

int main()

{

int n;

cin >> n;

vector<pair<float, float>> vec(n, make\_pair(0, 0));

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

{

cin >> vec[i].second;

}

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

{

cin >> vec[i].first;

vec[i].first = -1 \* vec[i].first / vec[i].second;

}

sort(vec.begin(), vec.end());

double capacity;

cin >> capacity;

double sum = 0;

for (auto i : vec)

{

if (i.second <= capacity)

{

sum += -1 \* i.first \* i.second;

capacity -= i.second;

}

else

{

sum += -1 \* i.first \* capacity;

capacity = 0;

break;

}

}

cout << sum << endl;

}