CSE246

Section 4

Fall 2023

**Lab Task - 02**

Submitted By

|  |  |
| --- | --- |
| **Name** | **ID** |
| B M Shahria Alam | 2021-3-60-016 |

**Problem 1**

**Solution:**

#include<bits/stdc++.h>

using namespace std;

using ll=long long;

int lowerBound(int a[],int key,int n)

{

int l=0,r=n-1,ans=0;

while(l<=r)

{

int mid =(l+r)/2;

if(a[mid]<=key)

{

l=mid+1;

ans=mid;

}

else r=mid-1;

}

return ans;

}

int main()

{

int n,k;

cin>>n>>k;

int a[n];

for(int i=0; i<n; i++)cin>>a[i];

int ans=0;

while(k!=0)

{

int lower=lowerBound(a,k,n);

ans=ans+(k/a[lower]);

k=k%a[lower];

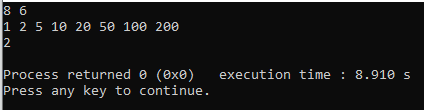
}

cout<<ans<<endl;

return 0;

}

**OUTPUT:**

****

**Problem 2**

**Solution:**

#include<bits/stdc++.h>

using namespace std;

struct knapsack

{

int value;

int weight;

};

void merge(knapsack arr[], int b, int mid, int e)

{

int n=mid-b+1;

int m=e-mid;

knapsack a[n];

knapsack bb[m];

for(int i=0; i<n; i++) a[i]=arr[b+i];

for(int i=0; i<m; i++) bb[i]=arr[mid+1+i];

int i=0;

int j=0;

int k=b;

while(i<n && j<m)

{

if(a[i].value<=bb[j].value)

{

arr[k]=a[i];

k++;

i++;

}

else

{

arr[k]=bb[j];

k++;

j++;

}

}

while(i<n)

{

arr[k]=a[i];

k++;

i++;

}

while(j<m)

{

arr[k]=bb[j];

k++;

j++;

}

}

void mergeSort(knapsack arr[], int b, int e)

{

if(b<e)

{

int mid = b+(e-b)/2;

mergeSort(arr,b,mid);

mergeSort(arr, mid+1, e);

merge(arr, b, mid, e);

}

}

int main()

{

int n;

cin>>n;

double w;

cin>>w;

knapsack a[100005];

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

{

int x;

cin>>x;

a[i].value=x;

}

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

{

int x;

cin>>x;

a[i].weight=x;

}

mergeSort(a,0,n-1);

int ans=0;

int i=n-1;

while(w!=0)

{

double min;

if(a[i].weight<w)min=a[i].weight;

else min=w;

w=w-min;

ans=ans+(min\*a[i].value);

i--;

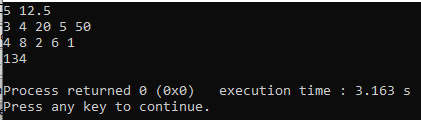
}

cout<<ans<<endl;

return 0;

}

**OUTPUT:**

****

**Problem 3**

**Solution:**

#include<bits/stdc++.h>

using namespace std;

struct work

{

int start;

int end;

};

void merge(work arr[], int b, int mid, int e)

{

int n = mid-b+1;

int m = e-mid;

work a[n];

work bb[m];

for(int i=0; i<n; i++) a[i]=arr[b+i];

for(int i=0; i<m; i++) bb[i]=arr[mid+1+i];

int i=0;

int j=0;

int k=b;

while(i<n && j<m)

{

if(a[i].end<=bb[j].end)

{

arr[k]=a[i];

k++;

i++;

}

else

{

arr[k]=bb[j];

k++;

j++;

}

}

while(i<n)

{

arr[k]=a[i];

k++;

i++;

}

while(j<m)

{

arr[k]=bb[j];

k++;

j++;

}

}

void mergeSort(work arr[], int b, int e)

{

if(b<e)

{

int mid = b+(e-b)/2;

mergeSort(arr,b,mid);

mergeSort(arr, mid+1, e);

merge(arr, b, mid, e);

}

}

int main()

{

int n;

cin>>n;

work a[n];

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

{

int n,m;

cin>>n>>m;

a[i].start=n;

a[i].end=m;

}

mergeSort(a,0,n-1);

int ans=1;

work b[n];

b[0].start=a[0].start;

b[0].end=a[0].end;

for(int i=0; i<n-1; i++)

{

if(b[i].end<=a[i+1].start)

{

ans++;

b[i+1].start=a[i+1].start;

b[i+1].end=a[i+1].end;

}

}

cout<<ans<<endl;

return 0;

}

**OUTPUT:**

