#include <stdio.h>

#define MAX 20

typedef struct

{

float w;

float p;

float r;

} Kobject;

int n;

float M;

void ReadObjects(Kobject obj[]);

void Knapsack(Kobject Kobj[]);

int main()

{

Kobject obj[MAX];

printf("Enter number of objects: ");

scanf("%d", &n);

ReadObjects(obj);

Knapsack(obj);

return 0;

}

void ReadObjects(Kobject obj[]) {

Kobject temp;

printf("Enter the max capacity of Knapsack: ");

scanf("%f", &M);

printf("Enter weight: ");

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

scanf("%f", &obj[i].w);

printf("Enter profits: ");

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

scanf("%f", &obj[i].p);

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

obj[i].r = obj[i].p / obj[i].w;

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

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

if(obj[j].r < obj[j+1].r) {

temp = obj[j];

obj[j] = obj[j+1];

obj[j+1] = temp;

}

}

void Knapsack(Kobject Kobj[]) {

float X[MAX];

float totalprofit = 0;

float u = M;

int i;

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

X[i] = 0;

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

if(Kobj[i].w > u)

break;

else {

X[i] = 1;

totalprofit += Kobj[i].p;

u -= Kobj[i].w;

}

}

printf("i=%d\n", i);

if(i < n) {

X[i] = u / Kobj[i].w;

totalprofit += (X[i] \* Kobj[i].p);

}

printf("The solution vector, X[]:\n");

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

printf("%.2f ", X[i]);

printf("\nTotal profit is: %.2f\n", totalprofit);

}

Output:

Enter number of objects: 7

Enter the max capacity of Knapsack: 15

Enter weight: 1 3 5 4 1 5 8

Enter profits: 5 10 15 8 3 9 4

i=5

The solution vector, X[]:

1.00 1.00 1.00 1.00 1.00 0.20 0.00

Total profit is: 42.80

=== Code Execution Successful ===