/*knapsack*/

```
#include<stdio.h>

int m=0,n=0;

void knapsack(float cal[],float p[],float w[]);

int main(){
	int i;
	float cal[20],p[20],w[20];
	printf("enter the max weight of knapsack: ");
	scanf("%d",&m);
	printf("\nenter the no. of objects: ");
	scanf("%d",&n);
	printf("\nenter weights & profits\n");
	for(i=0;i<n;i++){
		printf("w[%d]: ",i+1);
		scanf("%f",&w[i]);
		printf("p[%d]: ",i+1);
		scanf("%f",&p[i]);
	}
	for(i=0;i<n;i++){
		cal[i]=p[i]/w[i];
	}
	knapsack(cal,p,w);
	return 0;
}

void knapsack(float cal[],float p[],float w[]){
	int i,j;
	float temp;
```

```
float total_profit=0;
int selected_items[20];
int ct=0;
for(i=0;i<n;i++){
	for(j=i+1;j<n;j++){
		if(cal[i]<cal[j]){
			temp=cal[i];
			cal[i]=cal[j];
			cal[j]=temp;

			temp=p[i];
			p[i]=p[j];
			p[j]=temp;

			temp=w[i];
			w[i]=w[j];
			w[j]=temp;
		}
	}
}
printf("\n\nprofit\tweight\tcalc\n");
for(i=0;i<n;i++){
	printf("%.3f\t%.3f\t%.3f\n",p[i],w[i],cal[i]);
}

for(i=0;i<n;i++){
	if(m>0 && w[i]<=m){
		m-=w[i];
		total_profit+=p[i];
		selected_items[ct++]=i;
	}
```

```
                else{
                        break;
                }
        }

// If there is still space in the knapsack, add a fraction of the next item
 // if (m > 0 && i < n) {
// float fraction = (float)m / w[i];
 // total_profit += p[i] * fraction;
// selected_items[ct] = i;
// Store the index of the selected item
 // }

        printf("selected items are\n");
        for(i=0;i<ct;i++){
                printf("item_id: %d,weight: %.2f,profit:
%.2f\n",selected_items[i]+1,w[selected_items[i]],p[selected_items[i]]);
        }
        printf("total profit: %.2f\n",total_profit);
}
```