

第五回全統かんだ模試

以下のような問題を解くプログラムを考える。

1 円以上 1 0 0 0 円以下の金額がランダムに書かれた金券が 3 0 枚ある。
この 3 0 枚から金券がランダムに数枚（1 枚以上 3 0 枚以下）選ばれ、それらの合計金額を N とする。
入力を各金券に書かれている金額および合計金額 N として、合計金額が N となるような金券の選び方を求めるプログラムを作成せよ。

これを解くプログラムを図に示す。このプログラムをプログラム A とする。このプログラムにおいて、`data` が金券の金額を格納した整数型配列、`n` が合計金額 N として与えられているとする。なお、`data` は整列済みであるとする。

1. プログラム A のアルゴリズムについて説明した以下の文の空欄を適切に埋めよ。ただし、選択肢が用意されているものはその中から選べ。

まず配列 `sum` を定義する。これは、 i 枚目までの金券の合計金額が `sum[i]` に格納されているような配列である。ある目標金額 n について、全ての金券の合計金額よりも n の方が(①大きい・小さい)のであれば、合計金額 $-n$ となるような金券を選ぶほうが計算量が少ないと思われるからである。この判断をソースコード(②)行目の `if` 文で行っている。

関数 `select` では、与えられた目標金額を超えない範囲で金券を選ぶ処理が実装されている。まず、まだ選ばれていない金券の中から、与えられた目標金額を、(③超える・超えない)かつ目標金額に最も近い金券を探索する。これは関数 `search` によって実装され、そのとき用いられている探索方法は(④)である。もし見つからなければ、関数 `select` は -1 を返す。これは、金券の選択に失敗したことを示す。

見つかった場合、その金券の添字を i に格納する。このとき、添字が i (⑤以上・以下)のすべての金券の総額が目標金額に届かないのであれば、これ以上の解の探索は無意味であるので、 -1 を返す。

大丈夫であれば、 i 番目の金券を選び、残りの金券について再帰する。このとき、再帰した結果が成功であれば、呼び出した `select` 側もさらに 1 を返すが、失敗であれば、 i 番目の金券の選択を取りやめ、1 つ添字が(⑥大きな・小さな)金券を選びなおし、再帰する。これを成功するまで繰り返す。

もし残りのすべての金券について同様にしても失敗したならば、 -1 を返す。

2. 空欄 (ア) (イ) (ウ) を適切に埋めよ。
3. `data={4,7,10,16,21,30,38},n=65` のときについて以下の問に答えよ。
 - ① 配列 `sum` の内容を答えよ。
 - ② 関数 `select` が 1 回目に呼び出されたときの 21 行目において宣言された `i` の値を答えよ。
 - ③ 関数 `select` が 2 回目に呼び出されたときの 21 行目において宣言された `i` の値を答えよ。
 - ④ 関数 `select` が初めて返り値として `-1` を返すのは、関数 `select` が何回目に呼びだされたときか答えよ。また、そのとき、何行目の処理によって `-1` が返されたのかも答えよ。
 - ⑤ 正しい金券の選び方を答えよ。
4. この問題の作成にあたり、過去の授業で出題された問題を参考にした。その授業名を答えよ。

```

1  #include <stdio.h>
2
3  #define N 30
4  int answer[N]={};
5  int sum[N]={};
6
7  int search(int data[],int n,int l,int r){
8      if(l>=r){
9          if(data[l]>n) return -1;
10         else return l;
11     }
12     int m=(l+r)/2;
13     int ans1,ans2;
14     ans1=search(data,n,l,m);
15     ans2=search(data,n,m+1,r);
16     if( (ア) ) return ans1;
17     else return ans2;
18 }
19
20 int select(int data[],int n,int r){
21     int i=search(data,n,0,r);
22     if(i==-1) return -1;
23     if(sum[r]<n) return -1;
24     int k=i;
25     while(0<=k&&k<=i){
26         answer[k]++;
27         n-=data[k];
28         if(n==0) return 1;
29         int j=select( (イ) );
30         if(j>0) return 1;
31         answer[k]--;
32         n+=data[k];
33         k--;
34     }
35
36     if(k<0) return -1;
37 }
38 void main(){
39     /* ここでdataおよびnが与えられる*/
40     int i,flag=1;
41     sum[0]=data[0];
42     for(i=1;i<N;i++){
43         sum[i]=data[i]+sum[i-1];
44     }
45     if(n>sum[N-1]-n) {
46         n=sum[N-1]-n;
47         flag=0;
48     }
49     select(data,n,N-1);
50     for(i=0;i<N;i++){
51         if( (ウ) ) printf("%d¥n",i );
52     }
53 }

```