第４回日本情報オリンピック 本選問題 解答例

国際情報オリンピック日本委員会

1.(20 点)

```
#include <stdio.h>
#include <stdlib.h>

#define MAX 101

void main(void) {
    int size_of_seq, i, j, max_inc, max_dec, max_length=0, last, current;
    struct term {
        int value; /* 入力整数列の i 番目の項の値 */
        int inc;   /* i 番目の項で終わる最長の増加列の長さ */
        int inc_prev;
                   /* i 番目の項で終わる最長の増加列の直前の項のインデックス */
        int dec;   /* i 番目の項で終わる最長の減少列の長さ */
        int dec_prev;
                   /* i 番目の項で終わる最長の減少列の直前の項のインデックス */
    } seq[MAX];
    int solve[MAX];
    FILE *input_file, *output_file;

  /* 入力データの読み込み */
if ((input_file = fopen("INPUT.TXT", "r")) == NULL) {
printf("INPUT.TXT のオープンに失敗しました。\n");
        exit(1);
    }
    fscanf(input_file, "%d", &size_of_seq);
    for (i=1; i<=size_of_seq; i++)
          fscanf(input_file, "%d", &seq[i].value);
    fclose(input_file);

  /* i 番目の項で終わる最長の部分列の長さ（部分解）を求める */
    seq[1].inc = seq[1].dec =1;
    for (i=2; i<=size_of_seq; i++) {
        max_inc = max_dec = 0;
        for (j=1; j<=i; j++) {
          /* i-1 番目までの項に対する部分解が求まっているものとして、
             i 番目の項に対する部分解を求める */
            if (seq[j].value < seq[i].value && seq[j].inc > max_inc) {
                max_inc = seq[j].inc;
                seq[i].inc_prev = j;
            }
```

```
            if (seq[j].value > seq[i].value && seq[j].inc > max_dec) {
                max_dec = seq[j].dec;
                seq[i].dec_prev = j;
            }
        }
        seq[i].inc = max_inc+1;
        seq[i].dec = max_dec+1;
    }

  /* i 番目の項で終わる最長の部分列の中から最大のも（解）を見つけ、
     solve[] に格納 */
    for (i=1; i<=size_of_seq; i++) {
        if (seq[i].inc > max_length) {
            max_length=seq[i].inc; last=i;
        }
        if (seq[i].dec > max_length) {
            max_length=seq[i].dec; last=i;
        }
    }
    current = last;
    if (seq[last].inc > seq[last].dec) {
        for (i=max_length; i>=1; i--) {
            solve[i] = seq[current].value;
            current = seq[current].inc_prev;
        }
    } else {
        for (i=max_length; i>=1; i--) {
            solve[i] = seq[current].value;
            current = seq[current].dec_prev;
        }
    }

  /* 出力データの書き込み */
    if ((output_file = fopen("OUTPUT.TXT", "w")) == NULL) {
printf("OUTPUT.TXT のオープンに失敗しました。\n");
        exit(1);
    }
    fprintf(output_file, "%d\n", max_length);
    for (i=1; i<=max_length; i++)
        fprintf(output_file, "%d\n", solve[i]);
    fclose(output_file);
}
```

2.(20 点)

(1)

$$\left\{\frac{1}{2}, 1, 2, \ldots, 2^{n-3}, 2^{n-2}+\frac{1}{2}\right\}$$

(2)

```
#include <stdio.h>
#include <stdlib.h>

void main(void) {
    int n,j,s=1;
    long int i;
    FILE *input_file, *output_file;
        if ((input_file = fopen("INPUT.TXT", "r")) == NULL) {
            printf("INPUT.TXT のオープンに失敗しました。\n");
        exit(1);
    }
    fscanf(input_file, "%d%dl", &n, &i);
    fclose(input_file);
    if ((output_file = fopen("OUTPUT.TXT", "w")) == NULL) {
printf("OUTPUT.TXT のオープンに失敗しました。\n");
        exit(1);
    }
    s = s << n-1;
    if (i > 0) s--;
    i+=s;
    for (j=1;j<=n;j++) {
        if (i%2 == 1)
            fprintf(output_file,"1\n");
        else
            fprintf(output_file,"-1\n");
        i = i >> 1;
    }
    fclose(output_file);
}
```

3.(20 点)

```
#include <stdio.h>

#define SIZE 102
#define QUE 1000

enum jotai
{
  KABE='*', MICHI='.', ANSWER='+', LEFT=0, RIGHT, UP, DOWN
};

int que[QUE], top, bottom;
int meiro[SIZE][SIZE];

void push(int x, int y)
{
  que[bottom++]=x;
  que[bottom++]=y;
  bottom %= QUE;
}

void get(int *x, int *y)
{
  *x=que[top++];
  *y=que[top++];
  top %= QUE;
}

#define Is_Empty (top==bottom)

void solv(int sx, int sy, int gx, int gy)
{
  int i, nx, ny, cx, cy;
  int mv[4][3] = {{-1,0,RIGHT},{1,0,LEFT},{0,-1,DOWN},{0,1,UP}};
  cx=sx;
  cy=sy;
  push(cx,cy);
  do
    {
      get(&cx,&cy);
      for(i=0;i<4;i++)
{
  nx=cx+mv[i][0];
  ny=cy+mv[i][1];
  if(meiro[nx][ny]==MICHI)
    {
      meiro[nx][ny]=mv[i][2];
      push(nx,ny);
    }
```

```
}
    }
  while(((cx!=gx)||(cy!=gy))&&!Is_Empty);
  if((cx==gx)&&(cy==gy))
    {
      do
{
  nx=cx;
  ny=cy;
  cx += mv[meiro[nx][ny]][0];
  cy += mv[meiro[nx][ny]][1];
  meiro[nx][ny] = ANSWER;
}
      while((cx!=sx)||(cy!=sy));
      meiro[sx][sy] = ANSWER;
    }
}

void main()
{
  int wx, wy, sx, sy, gx, gy, i, j;
  char st[SIZE];
  FILE *fl;
  fl = fopen("INPUT.TXT","rt");
  fscanf(fl,"%d %d",&wx,&wy);
  for(i=0;i<=wx+1;i++)
    meiro[i][0] = meiro[i][wy+1] = KABE;
  for(j=0;j<=wy+1;j++)
    meiro[0][j] = meiro[wx+1][j] = KABE;
  fscanf(fl,"%d %d",&sx,&sy);
  fscanf(fl,"%d %d",&gx,&gy);
  for(j=1;j<=wy;j++)
    {
      fscanf(fl,"%s",st);
      for(i=1;i<=wx;i++)
meiro[i][j] = (int) st[i-1];
      fscanf(fl,"\n");
    }
  close(fl);
  solv(sx,sy,gx,gy);
  fl = fopen("OUTPUT.TXT","wt");
  for(j=1;j<=wy;j++)
    {
      for(i=1;i<=wx;i++)
switch(meiro[i][j])
  {
  case KABE: case ANSWER:
    fprintf(fl,"%c",(char) meiro[i][j]);
    break;
  default:
```

```
    fprintf(fl,"%c",MICHI);
  };
      fprintf(fl,"\n");
    }
  fclose(fl);
}
```

4.(20 点)

数列 $\{c_n\}, \{d_n\}$ をそれぞれ $c_n = a_{2^n}$, $d_n = b_{2^n}$ とおくと、 $N = 2^n$ $(n = 0, 1, 2, \ldots)$ のとき $a_{2N} = 2a_N + 4N$ より、

$$c_{n+1} = a_{2\cdot 2^n} = 2a_{2^n} + 4 \cdot 2^n$$

両辺を 2^{n+1} で割ると

$$\begin{aligned} \frac{c_{n+1}}{2^{n+1}} &= \frac{c_n}{2^n} + 2 \\ \therefore \quad \frac{c_n}{2^n} &= \frac{c_0}{2^0} + 2n = a_1 + 2n \end{aligned}$$

また、$b_{2N} = b_N + N^2$ より、

$$\begin{aligned} d_{n+1} &= b_{2\cdot 2^n} = b_{2^n} + (2^n)^2 \\ &= d_n + 4^n \\ \therefore \quad d_n &= d_0 + \sum_{i=0}^{n-1} 4^i = b_1 + \frac{4^n - 1}{4 - 1} \\ d_n &= b_1 + \frac{1}{3}(4^n - 1) \end{aligned}$$

自然数 n_0 を充分大きくすると 「$n > n_0$ ならば $2^n - 12n - 6a_1 - 12 > 1$」を満たすよう出来る。$N > 2^{n_0}$ のとき $2^{n+1} > N \geq 2^n$ となる n について

$$\begin{aligned} b_N - a_N &> b_{2^n} - a_{2^{n+1}} \\ &= b_1 + \frac{1}{3}(4^n - 1) - 2^{n+1}(a_1 + 2(n+1)) \\ &\geq \frac{1}{3}\left((2^n)^2 - 1 - 6 \cdot 2^n(a_1 + 2n + 2)\right) \\ &= \frac{1}{3}\left(2^n(2^n - 12n - 6a_1 - 12) - 1\right) \\ &> \frac{1}{3}(2^n - 1) \geq 0 \end{aligned}$$

従って、A 君のプログラムの方が効率がよい。