アルゴリズムとデータ構造 授業中練習問題５

次のプログラムは「スタックの実現例」である．このプログラムに関して，以下の問いに答えなさい．さらに，このプログラムを入力し，自分のパソコンでコンパイル，実行できることを確認してください．なお，プログラムの日本語部分は，英語，ローマ字に変更してかまいません．

```
#include <stdio.h>
#define MAX  10

/*--- int 型スタックを実現する構造体 ---*/
typedef struct {
  int max;      /* スタックの容量 */
  int ptr;      /* スタックポインタ */
  int stk[MAX]; /* スタック本体*/
} IntStack;

/*--- スタックの初期化 ---*/
int Initialize(IntStack *s, int max){
  s->ptr = 0;
  s->max = max;
  return 0;
}
/*--- スタックにデータをプッシュ ---*/
int Push(IntStack *s, int x){
  if (s->ptr >= s->max) return -1; /* スタック満杯 */
  s->stk[s->ptr] = x;
  s->ptr++;
  return 0;
}
/*--- スタックからデータをポップ ---*/
int Pop(IntStack *s, int *x){
  if (s->ptr <= 0) return -1; /* スタックは空 */
  s->ptr--;
  *x = s->stk[s->ptr];
  return (0);
}
/*--- スタックからデータをピーク ---*/
int Peek(IntStack *s, int *x){
  if (s->ptr <= 0) return -1;
  *x = s->stk[s->ptr - 1];
  return 0;
}
/*--- スタックの容量 ---*/
int Capacity(const IntStack *s){
  return s->max;
}
/*--- スタックに積まれているデータ数 ---*/
int Size(const IntStack *s){
  return s->ptr;
}
/*--- スタックの全データの表示 ---*/
void Print(const IntStack *s){
```

```
  int i;

  for(i = 0; i < s->ptr; i++)
    printf("%d ", s->stk[i]);
  putchar('¥n');
}

int main(void){
  IntStack  s;

  Initialize(&s, MAX);

  while (1) {
    int  menu, x;
    printf("現在のデータ数：%d/%d¥n",Size(&s), Capacity(&s));
    printf("(1) プッシュ  (2) ポップ  (3) ピーク  (4) 表示  (0) 終了：");
    scanf("%d", &menu);
                                                    ①
    if (menu == 0) break;

    switch (menu) {
     case 1: /* プッシュ */
       printf("データ：");
       scanf("%d", &x);
       if (Push(&s, x) == -1)
          puts("¥a エラー：プッシュに失敗しました。¥n");
       break;
     case 2: /* ポップ */
       if (Pop(&s, &x) == -1)
          puts("¥a エラー：ポップに失敗しました。¥n ");
       else
          printf("ポップしたデータは%d です。¥n", x);
       break;
     case 3: /* ピーク */
       if (Peek(&s, &x) == -1)
          puts("¥a エラー：ピークに失敗しました。¥n");
       else
          printf("ピークしたデータは%d です。¥n", x);
       break;
     case 4: /* 表示 */
       Print(&s);
       break;
     }
    }
    return 0;
}
```

1）このプログラムの動作直後に数字の２を入力しました．このとき，次の問に答えなさい．

(ア) スタックを操作するために **switch** 文中から呼び出される関数はどの関数ですか．

Pop(&s, &x)

(イ) (ア)で呼び出された関数の実行が終了し，**main** に戻った時のその関数の戻り値はいくらですか．

－１

(ウ) (ア)で呼び出された関数の実行が終了ときのスタックポインタの値は幾らですか．

０

2）スタックに「24」,「38」,「11」,「24」,「47」,「53」の順で値が積まれ，下線部①の部分でプログラムが入力待ちをしているとき，次の問に答えなさい．

(ア) **s.ptr** と **s.stk[2]**の値はいくらですか．

s.ptr: 6,　s.stk[2]: 11

(イ) ピークを指示した場合，ピークしたデータとして表示される値はいくらですか．

53

(ウ)(イ)でピークを指示し，ピークしたデータの値を表示した後，再度下線部①の部分でプログラムが入力待ちをしているときの **s.stk[s.ptr -2]**の値はいくらですか．

47

3）第３回講義中課題で用いた「身体検査データを定義する」以下の構造体 PhysCheck 型を，スタックで扱えるようにする「身体検査データ型スタックを実現する構造体」PhysCheckStack 型を定義し，「スタックの実現例」のプログラム中の IntStack 型を PhysCheckStack 型に変更しても正しく動作するようにしてください．以下では，PhysCheckStack 型の定義と，変更したプログラム中の Push 関数と Pop 関数を答えなさい．

```
/*--- 身体データ型 ---*/
typedef  struct{
  double  vision;   /* 視力 */
  int     height;   /* 身長 */
} Body ;
/*--- 身体検査データ型 ---*/
typedef  struct{
  Body   body;      /* 身体データ型 ---*/
  char   name[20]; /* 氏名 */
} PhysCheck ;
```

解答例

```
/*--- 身体検査データ型スタックを実現する構造体 ---*/
typedef struct {
  int          max;     /* スタックの容量 */
  int          ptr;      /* スタックポインタ */
  PhysCheck  stk[MAX]; /* スタック本体*/
}PhysCheckStack;

/*--- スタックにデータをプッシュ ---*/
int Push(PhysCheckStack *s, PhysCheck x){
  if (s->ptr >= s->max) return -1; /* スタック満杯 */
  s->stk[s->ptr] = x;
  s->ptr++;
  return 0;
}

/*--- スタックからデータをポップ ---*/
int Pop(PhysCheckStack *s, PhysCheck *x){
  if (s->ptr <= 0) return -1; /* スタックは空 */
  s->ptr--;
  *x = s->stk[s->ptr];
  return (0);
}
```