

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("エラー: プッシュに失敗しました。");
                break;
            case 2: /* ポップ */
                if (Pop(&s, &x) == -1)
                    puts("エラー: ポップに失敗しました。");
                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("エラー: ピークに失敗しました。");
                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。
 - (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。
 - (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。
- 2) スタックに「42」、「35」、「11」、「24」、「83」、「47」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[4]**の値はいくらですか。
 - (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。
 - (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 - 1]**の値はいくらですか。
- 3) 第 3 回講義資料の理解度確認問題で用いた「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。

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