

# プログラミング入門 第14回小テスト

学籍番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

解答時間 15分

## 問題0 (必須問題)

本日の授業の主な内容を3つ、箇条書きで述べなさい (各5点、合計15点)

(この問題を解答しないと他の問題は採点されません)

- 1.
- 2.
- 3.

解答はハンドアウトを参照

## 問題1

スタックとキューに関する次の文章について、空欄にあてはまる言葉を入れなさい。

(各5点 計30点) 注: /は前後どちらの答えでも良いことを示します

1. スタックにデータを挿入する操作を ( **プッシュ** / push )、データを取り出す操作を ( **ポップ** / pop ) と呼ぶ。スタックからデータを取り出す時は、( **最後** ) に入れたデータから順に取り出される。
2. キューにデータを格納すると、取り出す時には最初に入れたデータが初めに取り出される。このような方式は略して ( **F I F O** ) とも呼ばれる。
3. 整数データを格納できるスタックに、1, 3, 5, 7 というデータを順に格納した。このスタックから2つデータを取り出し、次に 2, さらに 4 というデータを格納すると、スタックの内容は順に ( **1** , **3** , **2** ), 4 となる。
4. 同様に 1, 3, 5, 7 というデータを順にキューに格納した場合、2つデータを取り出し、次に 2, さらに 4 というデータを格納すると、キューの内容は順に ( **5** , **7** , **2** ), 4 となる。

## 問題2

キューにデータを挿入する関数 enq を作成したい。この関数は挿入したい値 (int 型整数) 一つを引数とし、戻り値は無いものとする。データの配列と先頭位置の添字、最後尾の次の位置を表す添字は、それぞれ外部変数として

```
int queue[200]; /* データを格納する配列 */
int head = 0; /* データ先頭位置の添字 */
int tail = 0; /* データ最後尾の次の位置の添字 */
```

と宣言されている。enq 関数の機能は、新しいデータを挿入し、データの位置を表す添字を更新することであり、添字が配列の最後に達したら次は0に戻す。なお、ハンドアウトの例と違って、キューに空きがあるかのチェックは行わない。また、% 演算子は使用しない。このとき次のプログラムの空欄を埋めなさい。(各5点 計25点)

```
void enq( int data ) {
    queue[tail] = data; /* キューにデータ格納 */
    tail++;           /* 位置を更新 */
    if ( tail == 200 ) tail = 0; /* tail -= 200 でもよい */
}
```

### 問題3

A大学に通うBさんは、0.1から0.9まで0.1おきの数の2乗和、つまり $0.1^2 + 0.2^2 + \dots + 0.9^2$ を計算するつもりで、以下のようなプログラムを書いた。(紙面の節約のため、実行部分以外の変数宣言などは省略した。なおx, sumは浮動小数点型の変数)

```
sum = 0.0;
for ( x=0.1; x<1; x+=0.1 ) sum += x*x;
printf("%.2f\n", sum);
```

ところが正解は2.85なのに、実行結果はそれより1大きい3.85になってしまったという。この現象を説明する以下の文章の空欄に適切な語句を入れなさい。なお2番目の空欄\*には「1より大きい」「1に等しい」「1より小さい」のいずれかが入る。(各5点 計30点)

<説明>

実行結果から、Bさんのプログラムではforループの中が( **10** )回実行されたことが分かる。これは数値計算の誤差のせいで、最後にループの中に入った時には意図に反してxが( \* **1より小さい** )値になっていたためと考えられる。このような現象が起きるのは、計算機が数値を有限桁しか扱えないことで生じる、( **丸め誤差** )という誤差が原因である。0.1のような簡単な小数でも、計算機内部では( **二** )進数を使って表すので、有限桁に収まらないことがある。Bさんの場合、正しい結果を求めるにはループカウンタに( **整数型/int型など** )変数を用いるのが良い。数値計算で注意すべき誤差は他にもあり、大きさが極端に異なる数を加減算した時、小さい方の値が結果に反映されなくなってしまうことを( **情報落ち/積み残し** )といい、大きさがほぼ同じ値を引き算した時に有効数字の桁数が減ってしまうことを桁落ちという。注: /はどちらの用語でも良い