

# 前期末試験問題 (1E 情報処理基礎)

2006年12月06日

## 1 制御の流れ

[問1] リスト1~3のプログラムを実行した結果を [ア] ~ [コ] の中から選択せよ。

リスト 1: break .

```
#include <stdio.h>

int main(void)
{
    int i=0;

    while(i<=6){
        i++;
        printf("\ni=%d\t", i);
        if(4<=i)break;
        printf("ii=%d", i);
    }

    return 0;
}
```

リスト 2: continue

```
#include <stdio.h>

int main(void)
{
    int i=0;

    while(i<=6){
        i++;
        printf("\ni=%d\t", i);
        if(4<=i)continue;
        printf("ii=%d", i);
    }

    return 0;
}
```

リスト 3: goto

```
#include <stdio.h>

int main(void)
{
    int i=0;

    while(i<=6){
        i++;
        printf("\ni=%d\t", i);
        if(4<=i)goto next;
        printf("ii=%d", i);
    }

next:
    printf("ii=%d", i);

    return 0;
}
```

[ア]

```
i=1
i=2
i=3
```

[イ]

```
i=1
i=2
i=3
i=4
ii=5
ii=6
ii=7
```

[ウ]

```
ii=1
ii=2
ii=3
ii=4
i=5
i=6
i=7
```

[エ]

```
i=1    ii=1
i=2    ii=2
i=3    ii=3
i=4    ii=3
```

[オ]

```
i=1    ii=1
i=2    ii=2
i=3    ii=3
i=4    ii=4
```

[カ]

```
i=1
i=2
i=3
i=4    ii=4
i=5    ii=5
i=6    ii=6
i=7    ii=7
```

[キ]

```
i=1    ii=1
i=2    ii=2
i=3    ii=3
i=4
i=5
i=6
i=7
```

[ク]

```
i=4    ii=4
i=5    ii=5
i=6    ii=6
i=7    ii=7
```

[ケ]

```
i=1    ii=1
i=2    ii=2
i=3    ii=3
i=4    ii=4
```

[コ]

```
i=1    ii=1
i=2    ii=2
i=3    ii=3
i=4    ii=4
i=5    ii=5
i=6    ii=6
i=7    ii=7
```

## 2 関数と変数

### 2.1 関数と変数のスコープ

リスト 4 に関する問いである。

- [問 1] プログラム中の変数を (G) グローバル変数, (L) ローカル変数, (B) ブロック内宣言の変数に分けよ。解答欄の表に宣言している行と変数名, 該当する変数の種類 (G, L, B) を記述する。
- [問 2] 前問の変数の有効範囲を示せ。解答欄の表に有効範囲の行を記述する。
- [問 3] (1) グローバル変数, (2) ローカル変数, (3) ブロック内宣言宣言を優先度の高い順に並べよ。
- [問 4] プログラムでプロトタイプ宣言を行っている行を答えよ。
- [問 5] 実引数と仮引数を示せ。行と変数名を答える。
- [問 6] メイン関数がユーザー定義関数 hoge を呼び出したとき, 実引数と仮引数でどのような処理が行われるか?—答えよ。
- [問 7] ユーザー定義関数 hoge の戻り値の値となる変数名を答えよ。
- [問 8] ユーザー定義関数で, 引数や戻り値が無い場合の型を答えよ。
- [問 9] リスト 4 のプログラムの実行結果を示せ。

リスト 4: 関数の動作と変数に関する問いのプログラム。

```
1 #include <stdio.h>
2
3 int hoge(int a, int b);
4 int foo=100;
5 //=====
6 int main(void)
7 {
8     int foo, bar;
9
10    foo=10;
11
12    for (bar=1; bar<=3; bar++){
13        int j;
14        j=hoge(foo, bar);
15        printf("%d\n", j);
16    }
17
18    return 0;
19 }
20
21 //=====
22 int hoge(int a, int b)
23 {
24     int i;
25
26     i = a+b+foo;
27
28     return i;
29 }
```

## 2.2 記憶クラス

[問 1] 表 1 の [A]–[E] の記憶クラス指定子を答えよ。

[問 2] 表 1 の [a]–[e] 各記憶クラスの特徴に合致するものを下の選択肢 ア–オ から選べ。

[問 3] もし、変数宣言に記憶クラス指定子を省いたら、どの記憶クラスが使われるか？

表 1: 記憶クラスの特徴

記憶クラス	記憶クラス指定子	特徴
自動変数	[A]	[a]
ローカル変数の静的変数	[B]	[b]
グローバル変数の静的変数	[C]	[c]
外部変数	[D]	[d]
レジスター変数	[E]	[e]

### 選択肢

ア プログラムの実行に先立って、記憶領域の確保と初期が行われる。プログラムの実行中、記憶領域は保持される。この変数が使えるのは、宣言した関数のみ。

イ レジスターを記憶領域として使う。レジスターは CPU の記憶領域で、メインメモリーよりも高速のアクセスが可能である。したがって、処理を高速化できる。

ウ ローカル変数の場合は、関数が呼び出される毎に記憶領域の確保と初期化 (初期値の指定がある場合) が行われ、関数の処理が終わったときにその変数は消滅する。グローバル変数の場合は、プログラム実行中、変数の記憶領域は保持される。

エ 分割コンパイルの場合、そのファイルのみで使えるグローバル変数ということを示している。

オ これが指定されると、記憶領域の確保は行われず、どこか他のファイルのグローバル変数を使うことを表す。記憶領域は、どこか他のファイルで確保される。

### 3 プログラム作成

#### 3.1 関数の最大値を求める

関数

$$f(x) = -5x^2 - 6x + 10 - \sin x \quad -10 \leq x \leq 10$$

の最大値を計算するプログラムを作成せよ。ただし，条件は以下のとおりとする。

- 計算精度は，0.00001 とする。
- 関数  $f(x)$  の計算には，ユーザー定義関数を使うこと。

#### 3.2 三角形の面積と周長

リストは，三角形の面積と周長を計算するプログラムである。その動作は，以下のようになっている。

- 三角形の 3 辺の長さをキーボードから読み込む。これはメイン関数の役目。
- ひとつのユーザー定義関数で三角形の面積と周長を計算する。そのユーザー定義関数は，次のような動作を行う。

– 三辺の長さをそれぞれ， $a$  と  $b$ ， $c$  とすると面積  $S$  は，以下のヘロンの公式で計算する。

$$s = \frac{a + b + c}{2}$$
$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

- もし，あたえられた三辺で三角形ができないなら，ユーザー定義関数が計算する面積は負の値とする。
- ユーザー定義関数の結果にしたがい，メイン関数は，次の動作を行う。
  - 三角形ができなければ「入力した辺では三角形はできません」と表示する。
  - 三角形ができるならば，面積と周長を表示する。

## 4 応用問題

次の問題は難しい割には、配点が低い(各1点)。全ての問題が解け、見直しの完了した者のみトライせよ。

[問1] リスト5の実行結果を示せ。

[問2] リスト6の実行結果を示せ。

リスト5: 自分自身を呼び出す。

```
#include <stdio.h>

int f(int n);

int main(void)
{
    printf("%d\n", f(5));

    return 0;
}

int f(int n)
{
    if(n==0){
        return 1;
    }else{
        return n*f(n-1);
    }
}
```

リスト6: メイン関数を呼び出す。

```
#include <stdio.h>

int hoge(void);
int gv=3;

int main(void)
{
    printf("%d\n", hoge());

    return 0;
}

int hoge(void)
{
    if(gv!=0){
        gv--;
        main();
    }
    return gv++;
}
```