

1. 下に示したC言語のプログラムにおいて、行(X)から書かれている for ループでは、int 型配列 a に 0 から 99 までの int 型の乱数が 10 個格納される。この配列 a の内容にもとづいて、int 型配列 b の内容を次のように設定する処理を、プログラム中の枠内に記入せよ。すなわち、 $n = 0, 1, \dots, 9$  について、添え字  $n$  に対応する配列 b の要素  $b[n]$  が、配列 a の最初の要素  $a[0]$  から添え字  $n$  に対応する要素  $a[n]$  までの要素のうちの最大値となるように配列 b の内容を設定する処理を、プログラム中の枠内に記入せよ。ただし、配列 a の内容を変更してはいけない。また、プログラム中ですでに宣言されている変数以外の新たな変数を宣言してはいけない。

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

int main()
{
    int i;
    int a[10], b[10];

    for (i = 0; i < 10; i ++) {    // (X)
        a[i] = rand() % 100;
    }

```

```
        for (i = 0; i < 10; i ++) {
            printf("%d %d\n", i, b[i]);
        }
    return 0;
}
```

2. 下に示したC言語のプログラムにおいて、行(Y)から書かれている for ループでは、int 型配列 a に 0 から 9 までの int 型の乱数が 100 個格納される。この配列 a の内容にもとづいて、int 型配列 b の内容を次のように設定する処理を、プログラム中の枠内に記入せよ。すなわち、 $n = 0, 1, \dots, 9$  について、添え字  $n$  に対応する配列 b の要素  $b[n]$  が、配列 a の 100 個の要素のうち値が  $n$  以下である要素の個数となるように配列 b の内容を設定する処理を、プログラム中の枠内に記入せよ。ただし、配列 a の内容を変更してはいけない。また、プログラム中ですでに宣言されている変数以外の新たな変数を宣言してはいけない。

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

```
int main()
{
    int i;
    int a[100], b[10];

    for (i = 0; i < 100; i ++) { // (Y)
        a[i] = rand() % 10;
    }
```

```
        for (i = 0; i < 10; i ++) {
            printf("%d %d\n", i, b[i]);
        }
    return 0;
}
```

3. あなたは、次のC言語のプログラムを書き換えて、指定された出力を得るよう、依頼されている。

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

int main()
{
    int i, j;
    int m, n;
    int a[] = { 1, 2, 3, 4 }; // (P)
    int b[] = { 1, 2, 3, 4 }; // (Q)
    int c[4][4];

    for (i = 0; i < 4; i ++ ) {
        m = a[i];
        for (j = 0; j < 4; j ++ ) {
            n = b[j];
            c[i][j] = m + n; // (R)
        }
    }

    for (i = 0; i < 4; i ++ ) {
        for (j = 0; j < 4; j ++ ) {
            printf("%d ", c[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

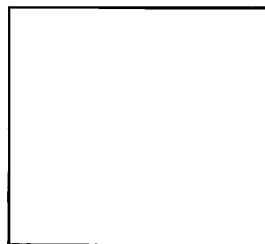
あなたが書き換えてもよいのは、行(P)、行(Q)、行(R)の3行だけである。しかも、行(P)と行(Q)については、`{ }`で囲まれた、要素数4の配列の初期値しか書き換えることができない。また、行(R)においては、代入記号 `=` の右側しか書き換えることができず、さらに、この行(R)を書き換える際に使えるのは、行末のセミコロン `;` を除いて、下に示す10種類の記号だけである。

m n 1 2 + - \* % ( )

これら10種類の記号であれば、行(R)の代入記号 `=` の右側において何回でも使ってよい。

このとき、以下の(1)から(4)までの問いに答えよ。

(1) あなたが書き換える前にこのプログラムを実行すると、どのような出力が得られるか、答えよ。



(2) 右のような出力を得るためには、プログラムをどのように書き換えればよいか、答えよ。

行(P)における配列の初期値	{ , , , };
行(Q)における配列の初期値	{ , , , };
行(R)における代入記号 = の右側	;

1 1 2 2
1 1 2 2
2 2 4 4
2 2 4 4

(3) 右のような出力を得るためには、プログラムをどのように書き換えればよいか、答えよ。

行(P)における配列の初期値	{ , , , };
行(Q)における配列の初期値	{ , , , };
行(R)における代入記号 = の右側	;

2 1 2 1
3 1 3 1
4 1 4 1
5 1 5 1

(4) 右のような出力を得るためには、プログラムをどのように書き換えればよいか、答えよ。

行(P)における配列の初期値	{ , , , };
行(Q)における配列の初期値	{ , , , };
行(R)における代入記号 = の右側	;

1 0 1 0
0 1 0 1
1 0 1 0
0 1 0 1

4. PC 上で広く使われている、マルチプログラミング、ページングによる仮想記憶、ラウンドロビンスケジューリングの機能を有する OS を対象とし、以下の設問 A~D に答えよ。解答は各問題文の下の枠中に記入すること。

A. OS は、各プロセスにページ（固定長に分割されたメモリ）を格納するページ枠を複数枠与える。そして、プロセスが新しく必要とするページが主記憶上に存在しない場合には、そのプロセスに与えた枠数を超えないようにページの置換えを行う。置換えアルゴリズムの 1 つとして、新しく必要になるページと次に使うまでの時間が最も長いページとを入れ替えるというものがある（未使用ページ枠があればそれを使用する）。これは **OPT** と呼ばれる。

ここで、あるプロセスにはメモリを固定長に分割したページを格納するページ枠が 3 枠与えられたとする。また、このプロセスは以下に示すページ使用系列（参照ストリング）を常に持つとする。**OPT** での置換え回数を答えよ。なお初期状態ではどのページ枠にもページが割当てられていないものとする。

0, 1, 2, 3, 1, 4, 0, 1, 2, 3, 4, 0

B. 前問の **OPT** は実現が困難であるため現実には使用されることはない。なぜ実現が困難なのか説明せよ。また、実現可能な、**OPT** の近似アルゴリズムにはどのようなものがあるか名称を挙げ、説明せよ。

C. プロセスには、実行状態、実行可能状態、待ち状態の 3 つがあるとす。プロセスが実行状態から実行可能状態に遷移するのはどのような場合か。

D. 複数のプログラムを実行した。主記憶の容量を半分に減らしてそれらの実行を再現すると、各プロセスの状態遷移数が増えた。なぜ増えたのか理由を挙げよ。

5. プロセスが実行中に用いるメモリをその使用目的によって、コード領域、データ領域、ヒープ領域、スタック領域の4領域に分類したとする。以下の設問 A～C に答えよ。解答は問題文の下の枠中に記入すること。

A. 定義される関数の種類や総行数を増やすとどの領域の大きさが増えるか。

B. グローバル変数の定義数を増やすとどの領域の大きさが増えるか。

C. スタック領域の大きさはプログラム実行直前には0バイトだが、実行を開始するとそれよりも増えている。なぜスタック領域はプログラムの実行中に増えるのか、説明せよ。

6. インターネット通信に関する以下の設問 A～C に答えよ。解答は問題文の下の枠中に記入すること。

A. IPv4 の IP パケットのヘッダ部では送信元や受信先を表すアドレスは何ビットで表されるか。

B. TCP パケットのヘッダ部には「シーケンス番号」と呼ばれるフィールドがあるが、UDP のパケットのヘッダ部には対応するものはない。なぜこのフィールドが TCP のヘッダ部には必要なのか説明せよ。

C. DNS とはどのような機能を提供するものか説明せよ。この機能が手元の PC で使えないとすると、どのようなことが起きるか一例を挙げよ。