

1. 下に示したC言語のプログラムについて（1）から（3）までの問いに答えなさい。なお、解答が複数ありうる場合は、そのうち一つを書けばよいものとする。

```
#include <stdio.h>

int main()
{
    int t[] = { 0, 0, 0, 0 };
    int b[] = { 1, 1, 1, -1 }; // (P)
    int i, c;

    i = 0;
    c = 0;
    for (;;) {
        t[c] += 1;
        if (b[i] == -1) { break; }
        c = (c + b[i]) % 4;
        i ++;
    }
    for (c = 0; c < 4; c ++ ) { printf("%d", t[c]); }
    printf("\n");

    return 0;
}
```

- (1) 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、下の枠内に書きなさい。

- (2) このプログラムを実行することで 2020 という出力を得たいとき、行(P)において配列 b[] をどのように初期化すればよいか、下の枠内に書きなさい。ただし、配列 b[] の要素の個数は自由に決めてよいものとする。

`int b[] = { _____ };`

- (3) このプログラムを実行することで 1232 という出力を得たいとき、行(P)において配列 b[] をどのように初期化すればよいか、下の枠内に書きなさい。ただし、配列 b[] の要素の個数は自由に決めてよいものとする。

`int b[] = { _____ };`

2. 下に示したC言語のプログラムについて（1）から（3）までの問いに答えなさい。なお、解答が複数ありうる場合は、そのうち一つを書けばよいものとする。

```
#include <stdio.h>

int main()
{
    int t[] = { 0, 0, 0, 0 };
    int b[] = { 0, 0, 0, -1 }; // (Q)
    int i, c;

    i = 0;
    c = 0;
    for (;;) {
        t[c] += 1;
        if (b[i] == -1) { break; }
        c = (c + t[b[i]]) % 4;
        i ++;
    }
    for (c = 0; c < 4; c ++) { printf("%d", t[c]); }
    printf("\n");

    return 0;
}
```

- (1) 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、下の枠内に書きなさい。

- (2) このプログラムを実行することで 2020 という出力を得たいとき、行(Q)において配列 b[] をどのように初期化すればよいか、下の枠内に書きなさい。ただし、配列 b[] の要素の個数は自由に決めてよいものとする。

```
int b[] = {                                     };
```

- (3) このプログラムを実行することで 1232 という出力を得たいとき、行(Q)において配列 b[] をどのように初期化すればよいか、下の枠内に書きなさい。ただし、配列 b[] の要素の個数は自由に決めてよいものとする。

```
int b[] = {                                     };
```

3. 下に示したC言語のプログラムについて（1）から（4）までの問いに答えなさい。なお、解答が複数ありうる場合は、そのうち一つを書けばよいものとする。

```
#include <stdio.h>

int s(int x[], int i, int j)
{
    int t;
    if (i < 0 || i > 5 || j < 0 || j > 5) { return -1; }
    t = x[i]; x[i] = x[j]; x[j] = t;
    return j;
}

int a(int n, int x[], int c)
{
    if (c == 0) { return s(x, n, n + 1); }
    else if (c == 1) { return s(x, n, n - 1); }
    else if (c == 2) { return s(x, n, n + 3); }
    else if (c == 3) { return s(x, n, n - 3); }
    return -1;
}

int main()
{
    int i;
    int z[] = { 1, 2, 3, 4, 5, 6 }; // (R)
    int y[] = { 3, 2, 1, 0, -1 }; // (S)
    int n;

    for (n = 0; n < 6; n++) { if (z[n] == 6) break; }
    i = 0;
    while (n != -1) { n = a(n, z, y[i++]); }

    for (n = 0; n < 6; n++) { printf("%d", z[n]); }
    printf("\n");

    return 0;
}
```

(1) 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、下の枠内に書きなさい。

(2) このプログラムを実行することで 612453 という出力を得たいとき、行(S)において配列 y[] をどのように初期化すればよいか、下の枠内に書きなさい。ただし、配列 y[] の要素の個数は自由に決めてよいものとする。

```
int y[] = { _____ };
```

(3) 行(R)を次のように書き換えたとする。

```
int z[] = { 4, 1, 3, 6, 2, 5 }; // (R)
```

このときに 123456 という出力を得たいならば、行(S)において配列 y[] をどのように初期化すればよいか、下の枠内に書きなさい。ただし、配列 y[] の要素の個数は自由に決めてよいものとする。

```
int y[] = { _____ };
```

(4) 行(R)を次のように書き換えたとする。

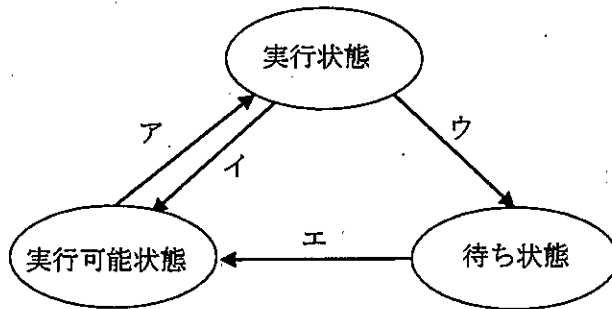
```
int z[] = { 6, 6, 0, 0, 0, 0 }; // (R)
```

このときに行(S)において配列 y[] を様々な値の列によって初期化することで得ることのできる出力のすべてを、下の枠内に重複がないように列挙しなさい。ただし、配列 y[] の要素の個数は自由に決めてよいものとする。

4. 以下の問いに答えよ。

- (1) 次の図は、オペレーティングシステムにおけるプロセスの三つの状態と、その遷移を示したものである。ア～エの状態間の遷移を引き起こす事象として適切なものを、それぞれ選択肢から選んで解答欄に記入せよ。

選択肢： イベント待ち, イベント発生, ディスパッチ, 再スケジュール



解答欄

| | |
|---|--|
| ア | |
| イ | |
| ウ | |
| エ | |

- (2) タイムスライスを導入してプロセスのスケジューリングを行う方法として、ラウンドロビンが知られている。下の表のように、プロセス A, B, C が異なる時刻に順次生成されたとする。各プロセスは表に示した処理時間を要するとすると、20 ms（ミリ秒）のタイムスライスでラウンドロビンによりスケジューリングを行ったとき、プロセス A, B, C はそれぞれのプロセスが生成されてから何ミリ秒後に処理が完了するか。解答欄に記入せよ。

| プロセス | 生成時刻(ms) | 処理時間(ms) |
|------|----------|----------|
| A | 0 | 30 |
| B | 5 | 25 |
| C | 10 | 10 |

解答欄

| | |
|---|--|
| A | |
| B | |
| C | |

- (3) ページングによる仮想記憶を考える。ページの置き換えを行う LRU アルゴリズムとはどのようなものかを説明せよ。

解答欄

- (4) あるプロセスにより参照されるページ番号が、以下であったとする。

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0 \rightarrow 1 \rightarrow 4 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

主記憶のページ枠が 4 であるとき、LRU アルゴリズムを用いるとページフォールトが何回発生するか説明せよ。ただし初期状態では主記憶にどのページも読み込まれていないとする。

解答欄