

SSC0600 - Introdução à Ciência de Computação I  
 Tópico: Estruturas de Repetição e Tipos de Dados  
 Compostos (Strings e Vetores)

Provinha 2(a)

21 de abril de 2017

N.º USP:

<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0
<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1
<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2
<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3
<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4
<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5
<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6
<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7
<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8
<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9

← Por favor codifique seu Número USP na esquerda e escreva seu nome abaixo.

Nome e sobrenome:
.....
.....

**Question 1 ♣** Marque (X) nas opções que são usadas para representar as **estruturas de repetição (loops)** em pseudocódigo ou na Linguagem C.

- for (*expressão*; *expressão*; *expressão*) { ... }
- ENQUANTO *expressão* FAÇA ... EMOUTROSCASOSFAÇA ... FIMENQUANTO
- SENÃO *expressão* ENTÃO ... SENÃO ...
- if-not (*expressão*) { ... } else { ... }
- SE *expressão* ENTÃO ... SENÃO ...
- do { ... } until (*expressão*) { ... }
- PARA *expressão* ATÉ *expressão* PASSO *expressão* FAÇA ... FIMPARA
- if (*expressão*) { ... }
- if-not (*expressão*) { ... }
- if (*expressão*) { ... } else { ... }
- PARA *expressão* ATÉ *expressão* FAÇA ... FIMPARA
- ENQUANTO *expressão* FAÇA ... FIMENQUANTO
- REPITA *expressão* ATÉ *expressão*
- while (*expressão*) { ... }
- SE *expressão* ENTÃO ...
- do { ... } while (*expressão*);
- while (*expressão*) { ... } other-case { ... }
- SENÃO *expressão* ENTÃO ...
- Nenhuma das alternativas está correta

**Question 2 ♣** Em relação aos trechos de código em Linguagem C apresentados nas Listagem 1, 2, 3 e 4; Marque (X) nas afirmativas verdadeiras do propósito de cada um.

- *Observação:* `arr[n][n]` é a matriz  $n \times n$  de números inteiros, matriz na que os índices para acessar a um valor vão de 0 a  $n-1$ .

- Listagem 4 gira 270 graus no sentido anti-horário a matriz `arr[n][n]`
- Listagem 4 gira 90 graus no sentido horário a matriz `arr[n][n]`
- Listagem 2 gira 90 graus no sentido horário a matriz `arr[n][n]`
- Listagem 4 gira 90 graus no sentido anti-horário a matriz `arr[n][n]`
- Listagem 2 gira 360 graus no sentido horário ou antihorário a matriz `arr[n][n]`
- Listagem 3 gira 360 graus no sentido horário ou antihorário a matriz `arr[n][n]`
- Listagem 1 gira 90 graus no sentido horário a matriz `arr[n][n]`
- Listagem 1 gira 180 graus no sentido horário ou antihorário a matriz `arr[n][n]`
- Listagem 3 gira 90 graus no sentido anti-horário a matriz `arr[n][n]`
- Listagem 2 gira 90 graus no sentido anti-horário a matriz `arr[n][n]`
- Listagem 1 gira 360 graus no sentido horário ou antihorário a matriz `arr[n][n]`
- Listagem 2 gira 270 graus no sentido horário a matriz `arr[n][n]`
- Listagem 4 gira 360 graus no sentido horário ou antihorário a matriz `arr[n][n]`
- Nenhuma das alternativas está correta*

**Question 3 ♣** Marque (X) na saída do programa apresentado na Listagem 6.

- |                          |                         |                          |                         |                                     |                         |                          |                         |
|--------------------------|-------------------------|--------------------------|-------------------------|-------------------------------------|-------------------------|--------------------------|-------------------------|
| <input type="checkbox"/> | 2 2 2<br>2 2 2<br>2 2 2 | <input type="checkbox"/> | 0 0 0<br>1 1 1<br>2 2 2 | <input checked="" type="checkbox"/> | 0 1 2<br>0 1 2<br>0 1 2 | <input type="checkbox"/> | 2 1 0<br>2 1 0<br>2 1 0 |
|--------------------------|-------------------------|--------------------------|-------------------------|-------------------------------------|-------------------------|--------------------------|-------------------------|
- Nenhuma das alternativas está correta*

**Question 4 ♣** Marque (X) nas afirmativas verdadeiras em relação ao programa da Listagem 7.

- Depois que o código for executado,  $x$  contém os valores: {10, 8, 2, 3, 1, 0}
- Depois que o código for executado,  $x$  contém os valores: {5, 4, 2, 3, 2, 0}
- Depois que o código for executado,  $y$  contém os valores: {5, 2, 6, 4, 4, 0}
- Depois que o código for executado,  $y$  contém os valores: {5, 1, 2, 6, 8, 0}
- Depois que o código for executado,  $y$  contém os valores: {5, 2, 4, 3, 4, 0}
- Depois que o código for executado,  $x$  contém os valores: {5, 4, 3, 4, 2, 0}
- Depois que o código for executado,  $y$  contém os valores: {5, 1, 6, 4, 8, 0}
- Depois que o código for executado,  $x$  contém os valores: {10, 8, 6, 2, 1, 0}
- Nenhuma das alternativas está correta*

**Question 5** ☒ Marque (X) nas afirmativas verdadeiras em relação ao programa da Listagem 7.

- A condição  $i1 < j1$  da Linha 9 é avaliada 3 vezes
- O código na estrutura de repetição interna (*inner loop*), linhas 17 até 21, é repetido 4 vezes
- O código na estrutura de repetição externa (*outer loop*), linhas 10 até 25, é repetido 6 vezes
- A condição  $j2 > i2$  da Linha 16 é avaliada 4 vezes
- A condição  $i1 < j1$  da Linha 9 é avaliada 4 vezes
- O código na estrutura de repetição externa (*outer loop*), linhas 10 até 25, é repetido 3 vezes
- O código na estrutura de repetição interna (*inner loop*), linhas 17 até 21, é repetido 6 vezes
- O código na estrutura de repetição externa (*outer loop*), linhas 10 até 25, é repetido 4 vezes
- A condição  $j2 > i2$  da Linha 16 é avaliada 6 vezes
- A condição  $j2 > i2$  da Linha 16 é avaliada 3 vezes
- A condição  $i1 < j1$  da Linha 9 é avaliada 6 vezes
- O código na estrutura de repetição interna (*inner loop*), linhas 17 até 21, é repetido 3 vezes
- Nenhuma das alternativas está correta

**Question 6** ☒ Marque (X) nas modificações que, de maneira independente umas das outras, conseguem que o programa apresentado na Listagem 6 imprima:

```
0 0 0
1 1 1
2 2 2
```

- A linha 12 deve ser mudada para: `for (i=2; i>=0; i-) {`  
A linha 14 deve ser mudada para: `printf("%d ", m[j][i]);`
- A linha 12 deve ser mudada para: `for (i=2; i>=0; i-) {`
- A linha 9 deve ser mudada para: `m[col][row] = col;`
- A linha 14 deve ser mudada para: `printf("%d ", m[j][i]);`
- A linha 13 deve ser mudada para: `for (j=2; j>=0; j-) {`  
A linha 14 deve ser mudada para: `14 printf("%d ", m[j][i]);`
- A linha 9 deve ser mudada para: `m[row][col] = row;`
- A linha 13 deve ser mudada para: `for (j=2; j>=0; j-) {`
- Nenhuma das alternativas está correta

**Question 7** ☒ O trecho de código apresentado na Listagem 5 tem sido proposto para efetuar a ordenação ascendente (de menor a maior) de um vetor de inteiros `arr[n]` (array `arr` de tamanho `n`). No entanto, o programa não funciona adequadamente. Marque (X) nas modificações necessárias que, em conjunto, conseguem fazer o código funcionar adequadamente.

- A linha 5 deve ser mudada para: `if (arr[j] > arr[m]) {`
- A linha 4 deve ser mudada para: `while (j < n) {`
- A linha 4 deve ser mudada para: `while (j >= n-1) {`
- A linha 4 deve ser mudada para: `while (j < n-1) {`
- A linha 4 deve ser mudada para: `while (j > n) {`
- A linha 5 deve ser mudada para: `if (arr[j] >= arr[m]) {`
- A linha 5 deve ser mudada para: `if (arr[j] < arr[m]) {`
- Nenhuma das alternativas está correta

---

```
1 for(j=n-1; j>=0; j--) {
2     for(k=n-1; k>=0; k--) {
3         printf("%d ", arr[j][k]);
4     }
5     printf("\n");
6 }
```

---

Listagem 1: Trecho de código para girar uma matriz  $n \times n$

---

```
1 for (j=0; j<n; j++) {
2     for(k=n-1; k>=0; k--) {
3         printf("%d ", arr[k][j]);
4     }
5     printf("\n");
6 }
```

---

Listagem 2: Trecho de código para girar uma matriz  $n \times n$

---

```
1 for(j=0; j<n; j++) {
2     for(k=0; k<n; k++){
3         printf("%d ", arr[j][k]);
4     }
5     printf("\n");
6 }
```

---

Listagem 3: Trecho de código para girar uma matriz  $n \times n$

---

```
1 for(j=n-1; j>=0; j--) {
2     for(k=0; k<n; k++){
3         printf("%d ", arr[k][j]);
4     }
5     printf("\n");
6 }
```

---

Listagem 4: Trecho de código para girar uma matriz  $n \times n$

---

```
1 for (i = 0; i < n-1; i++) {
2     m = i;
3     j = i+1;
4     while (n-1 > j) {
5         if (arr[m] < arr[j]) {
6             m = j;
7         }
8         j++;
9     }
10
11     if (i != m) {
12         aux = arr[i];
13         arr[i] = arr[m];
14         arr[m] = aux;
15     }
16 }
```

---

Listagem 5: Trecho de código para ordenar um vetor de inteiros  $arr[n]$

---

---

```

1 #include <stdio.h>
2
3 int row, col, i, j;
4 int m[3][3];
5
6 int main() {
7     for (row = 0; row <= 2; row++) {
8         for (col = 0; col <= 2; col++) {
9             m[row][col] = col;
10        }
11    }
12    for (i=0; i<3; i++) {
13        for (j=0; j<3; j++) {
14            printf("%d ", m[i][j]);
15        }
16        printf("\n");
17    }
18    return 0;
19 }

```

---

Listagem 6: Código de programa na linguagem C

---

```

1 #include <stdio.h>
2
3 int x[6] = {0, 1, 2, 3, 4, 5};
4 int y[6] = {5, 4, 3, 2, 1, 0};
5 int i1 = 0, j1 = 5, i2 = 0, j2 = 5;
6 int temp;
7
8 int main() {
9     while (i1 < j1) {
10        temp = x[i1];
11        x[i1] = x[j1];
12        x[j1] = 2*temp;
13
14        i2 = i1+1;
15        j2 = j1-1;
16        while (j2 > i2) {
17            temp = y[j2];
18            y[j2] = y[i2];
19            y[i2] = 2*temp;
20            i2++;
21            j2--;
22        }
23
24        i1++;
25        j1--;
26    }
27    return 0;
28 }

```

---

Listagem 7: Código de programa na linguagem C