

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

Provinha 2(b)

8 de maio 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"/>	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"/>	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"/>	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"/>	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"/>	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"/>	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"/>	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"/>	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"/>	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	<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.

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

Question 2 ♣ Sejam os quatro tipos de pirâmides: (a) pirâmide completa, (b) pirâmide completa invertida, (c) meia pirâmide e (d) meia pirâmide invertida (Figura 1), Marque (X) nas afirmativas verdadeiras em relação as Listagem 1, 2, 3 e 4.

- *Observação:* n é a altura (ou número de filas) das pirâmides

- Listagem 2 imprime uma pirâmide completa
- Listagem 3 imprime uma meia pirâmide invertida
- Listagem 2 imprime uma pirâmide completa invertida
- Listagem 1 imprime uma pirâmide completa
- Listagem 3 imprime uma pirâmide completa invertida
- Listagem 2 imprime uma meia pirâmide invertida
- Listagem 1 imprime uma meia pirâmide
- Listagem 3 imprime uma meia pirâmide
- Listagem 2 imprime uma meia pirâmide
- Listagem 4 imprime uma pirâmide completa
- Listagem 4 imprime uma meia pirâmide
- Listagem 4 imprime uma pirâmide completa invertida
- Listagem 3 imprime uma pirâmide completa
- Listagem 1 imprime uma meia pirâmide invertida
- Listagem 1 imprime uma pirâmide completa invertida
- Listagem 4 imprime uma meia pirâmide invertida
- Nenhuma das alternativas está correta*

Question 3 ♣ Marque (X) na saída do programa apresentado na Listagem 5.

- | | | | | | | | |
|--------------------------|-------------------------|--------------------------|-------------------------|-------------------------------------|-------------------------|--------------------------|-------------------------|
| <input type="checkbox"/> | 0 1 2
1 2 3
2 3 4 | <input type="checkbox"/> | 0 1 2
3 4 5
6 7 8 | <input checked="" type="checkbox"/> | 0 3 6
1 4 7
2 5 8 | <input type="checkbox"/> | 0 3 6
1 4 5
2 7 8 |
|--------------------------|-------------------------|--------------------------|-------------------------|-------------------------------------|-------------------------|--------------------------|-------------------------|
- Nenhuma das alternativas está correta*

Question 4 ♣ Marque (X) nas afirmativas verdadeiras em relação ao programa da Listagem 6.

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

Question 5 ♣ Marque (X) nas afirmativas verdadeiras em relação ao programa da Listagem 6.

- A condição $i1 > 0$ da Linha 10 é avaliada 3 vezes
- O código na estrutura de repetição interna (*inner loop*), linhas 18 até 22, é executado 4 vezes
- A condição $i2 < j2$ da Linha 17 é avaliada 5 vezes
- O código na estrutura de repetição interna (*inner loop*), linhas 18 até 22, é executado 5 vezes
- A condição $i2 < j2$ da Linha 17 é avaliada 7 vezes
- A condição $i1 > 0$ da Linha 10 é avaliada 4 vezes
- A condição $i1 > 0$ da Linha 10 é avaliada 2 vezes
- O código na estrutura de repetição interna (*inner loop*), linhas 18 até 22, é executado 6 vezes
- O código na estrutura de repetição externa (*outer loop*), linhas 11 até 26, é executado 4 vezes
- O código na estrutura de repetição externa (*outer loop*), linhas 11 até 26, é executado 2 vezes
- O código na estrutura de repetição externa (*outer loop*), linhas 11 até 26, é executado 3 vezes
- A condição $i2 < j2$ da Linha 17 é avaliada 6 vezes
- Nenhuma das alternativas está correta

Question 6 ♣ Marque (X) nas modificações que, de maneira independente umas das outras, façam com que o programa apresentado na Listagem 5 imprima:

```
8 5 2
7 4 1
6 3 0
```

- A linha 10 deve ser mudada para: $m[2-j][2-i] = k;$
- A linha 10 deve ser mudada para: $m[2-i][2-j] = k;$
- A linha 10 deve ser mudada para: $m[j][i] = k;$
e a linha 13 deve ser mudada para: `for (i=2; i >= 0; i--) {`
e a linha 14 deve ser mudada para: `for (j=2; j >= 0; j--) {`
e a linha 15 deve ser mudada para: `printf("%d ", m[j][i]);`
- A linha 15 deve ser mudada para: `printf("%d ", m[j][i]);`
- A linha 13 deve ser mudada para: `for (i=2; i >= 0; i--) {`
e a linha 14 deve ser mudada para: `for (j=2; j >= 0; j--) {`
- A linha 10 deve ser mudada para: $m[j][i] = k;$
- A linha 14 deve ser mudada para: `for (j=2; j >= 0; j--) {`
- A linha 13 deve ser mudada para: `for (i=2; i >= 0; i--) {`
- Nenhuma das alternativas está correta

Question 7 ♣ O trecho de código apresentado na Listagem 7 tem sido proposto para efetuar a ordenação descendente (de maior a menor) de um vetor de inteiros $arr[n]$ de tamanho n . No entanto, o código não funciona adequadamente. Marque (X) nas modificações necessárias que, em conjunto, façam o programa funcionar adequadamente.

- A linha 1 deve ser mudada para: `for (i = 1 ; i < n-1 ; i++) {`
- A linha 3 deve ser mudada para: `while (j > 0 && arr[j] < arr[j-1]) {`
- A linha 3 deve ser mudada para: `while (j > 0 && arr[j] > arr[j-1]) {`
- A linha 3 deve ser mudada para: `while (j > i && arr[j] > arr[j-1]) {`
- A linha 2 deve ser mudada para: $j = i-1;$
- A linha 1 deve ser mudada para: `for (i = n-2 ; i >= 0; i--) {`
- A linha 2 deve ser mudada para: $j = i+1;$
- A linha 1 deve ser mudada para: `for (i = 0 ; i < n-1 ; i++) {`
- Nenhuma das alternativas está correta

CORRECTED

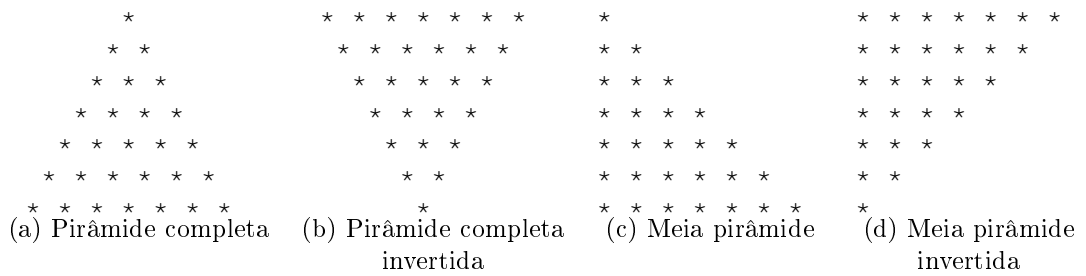


Figura 1: Exemplos de tipos de pirâmides com altura $n=7$

```

1 for (i=0 ; i < n ; i++) {
2     for (j=0 ; j <= i ; j++) {
3         printf("* ");
4     }
5     printf("\n");
6 }

```

Listagem 1: Trecho de código para imprimir uma pirâmide

```

1 for (i=0; i < n; i++) {
2     for (k=0; k < i; k++) {
3         printf(" ");
4     }
5     for (k=0 ; k < n-i; k++) {
6         printf("* ");
7     }
8     printf("\n");
9 }

```

Listagem 2: Trecho de código para imprimir uma pirâmide

```

1 for (i=n ; i > 0 ; i--) {
2     for (j=0 ; j < i ; j++) {
3         printf("* ");
4     }
5     printf("\n");
6 }

```

Listagem 3: Trecho de código para imprimir uma pirâmide

```

1 for (i=0; i < n; i++) {
2     for (k=0; k < (n-i)-1; k++) {
3         printf(" ");
4     }
5     for (k=0 ; k < i+1; k++) {
6         printf("* ");
7     }
8     printf("\n");
9 }

```

Listagem 4: Trecho de código para imprimir uma pirâmide

```

1 #include <stdio.h>
2
3 int m[3][3];
4 int row, col, i, j, k;
5
6 int main() {
7     for (k = 0; k < 9; k++) {
8         i = k % 3;
9         j = k / 3;
10        m[i][j] = k;
11    }
12
13    for (i=0; i < 3; i++) {
14        for (j=0; j < 3; j++) {
15            printf("%d ", m[i][j]);
16        }
17        printf("\n");
18    }
19    return 0;
20 }

```

Listagem 5: Código de programa na linguagem C

```

1 #include <stdio.h>
2
3 int x[6] = {5, 4, 3, 2, 1, 0};
4 int y[6] = {0, 1, 2, 3, 4, 5};
5
6 int i1 = 2, j1 = 3;
7 int i2, j2, temp;
8
9 int main() {
10    while (i1 > 0) {
11        temp = x[i1];
12        x[i1] = x[j1] * 2;
13        x[j1] = temp;
14
15        i2 = i1 - 1;
16        j2 = j1 + 1;
17        while (i2 < j2) {
18            temp = y[j2];
19            y[j2] = y[i2];
20            y[i2] = 2 * temp;
21            i2++;
22            j2--;
23        }
24
25        i1--;
26        j1++;
27    }
28    return 0;
29 }

```

Listagem 6: Código de programa na linguagem C

```
1 for (i = 1 ; i < n-1 ; i++) {
2     j = i;
3     while (j > 0 && arr[j-1] > arr[j]) {
4         aux = arr[j];
5         arr[j] = arr[j-1];
6         arr[j-1] = aux;
7         j--;
8     }
9 }
```

Listagem 7: Trecho de código para ordenar um vetor de inteiros `arr[n]`