

1 ESTADO E ATRIBUIÇÕES

①
 $x: 8 \ 12 \rightarrow$ o output vai ser: 12 16
 $y: 4 \ 16$

②
 $x: 0 \rightarrow$ o output vai ser: 0 (?)
 $y: ?$
 mº qualquer

NOTA: mas se colocarmos o programa no link dado pelo professor, vai dar erro.

③
 $a: 65 \ 66$
 $b: 32$
 $c: 48 \ 50 \ 98$
 \rightarrow o output vai ser:
 'A' 65
 'B' 66 '2' 50
 'b' 98

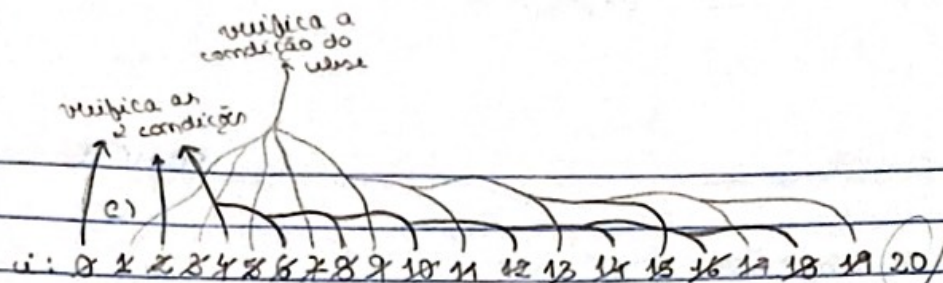
④
 $x: 200 \ 300 \ 100 \rightarrow$ o output vai ser: 100 200
 $y: 100 \ 200$

2 ESTRUTURAS DE CONTROLO

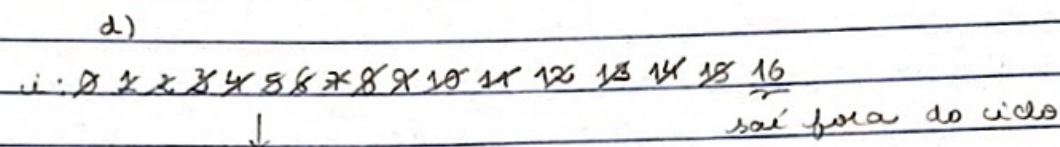
①
 a)
 $x: 3$
 $y: 5$
 ~~~~~  
 ã se verifica  
 a condição  $x > y$

aparentemente este último passo tem conta.

b)  
 $x: 8 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \rightarrow$  o output vai ser:  
 $y: 8 \ 1 \ 3 \ 6 \ 10 \ 15 \ 21 \ 28 \ 36 \ 45 \ 55 \ 66$   
 ~~~~~  
 verifica
 $x! = 11$
 deixou de verificar a condição



o output vai usar:
 - * - * - * - * - * - * - *
 quando o i chega ao 20, é criando o que entra dentro do ciclo "for".



o output vai usar:

'm' → m é o quanto aparece cada (as mudanças de linha)

'1' → como '1' é o número máximo de vezes do ciclo

'0' '1'

'1' '1'

'0' '0' '1'

'1' '0' '1'

'0' '1' '1'

'1' '1' '1'

'0' '0' '0' '1'

'1' '0' '0' '1'

'0' '1' '0' '1'

'1' '1' '0' '1'

'0' '0' '1' '1'

'1' '0' '1' '1'

'0' '1' '1' '1'

'1' '1' '1' '1'

(%) → resto da divisão inteira.

(/) → divisão inteira.

3 PROGRAMAS ITERATIVOS

①

argumentos = 5

```

# # # # #
# # # # #
# # # # #
# # # # #
# # # # #
  
```

imagens que form
 que aparece no
 output


```
#include <stdio.h>
void desenhaImagem (int m);
int main() {
    int c;
    scanf ("%d", &c);
    desenhaImagem (c);
    return 0;
}
```

putchar

↳ escreve um char na tela

```
void desenhaImagem (int m) {
    int l, c;
    for (l=0; l<m; l++) {
        for (c=0; c<m; c++) {
            putchar ('#');
            putchar ('\n');
        }
    }
}
```

→ neste tipo de questão tem que usar sempre 1º na linha e depois na coluna !! (ou vice-versa).

②

argumento = 5

④ → quando é par

① → quando é ímpar

0 1 2 3 4

```
# - # - #
- # - # -
# - # - #
- # - # -
# - # - #
```

imagem que tem que aparecer no output

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

```
void desenhaImagem (int m);
```

```
int main() {
```

```
    int c;
```

```
    scanf ("%d", &c);
```

```
    desenhaImagem (c);
```

```
    return 0;
```

```
}
```



```

void demonstrate (int m) {
    int x=0;
    int c, d;
    for (d=0; d<m; d++) {
        for (c=x; c<m+x; c++) {
            if (c%2==0) { putchar("#"); }
            else { putchar("-"); }
        }
        putchar("\n");
    }
}

```

③

argument = 5

vertical:

```

#
##
###
####
#####
#####
#####
####
###
##
#

```

horizontal:

```

      ① # 1
      ③ ## 2
      ⑤ ### 3
      ⑦ #### 4
      ⑨ ##### 5

```

```

void mputchar (int m, char c);

```

```

void vertical (int m) {

```

```

    int i;

```

```

    for (i=1; i<=m; i++) {

```

```

        mputchar (i, '#');

```

```

        putchar ('\n');
    }

```

```

    for (i=m; i>0; i--) {

```

```

        mputchar (i, '#');

```

```

        putchar ('\n');
    }
}

```

```

void horizontal (int m) {
    int i, l;
    i = 1;
    for (l = 1; l <= m; l++) {
        putchar (i, '#');
        putchar ('\n');
        i = i + 2;
    }
}

```

```

void mputchar (int m, char c) {
    int i;
    for (i = 0; i < m; i++) {
        putchar (c);
    }
}

```

④ (Resolução da aula!)

argumento \equiv raio = 4

```

void circulo (int x) {
    int x, y;
    for (y = -x; y <= x; y++) {
        for (x = -x; x <= x; x++) {
            if (x*x + y*y <= x*x) {
                putchar ('#'); putchar ('#'); }
            else { putchar (' '); putchar (' '); }
        }
    }
    putchar ('\n');
}

```