

- *Observações:*
- *-Coloquem a resolução de vcs e indiquem seu nome com @SeuNome em uma cor específica*
- *-Tentem colaborar colocando suas resoluções, não só esperando os amiguinhos resolverem (:*

Observações gerais: Você receberá folhas de almaço onde deve apresentar as respostas. Estas podem ser escritas a lápis e podem ser dadas fora de ordem, desde que corretamente indicadas. As folhas usadas como rascunho, caso você julgue necessário, devem ser entregues junto com a avaliação. Todas as folhas que você quiser entregar deverão ter seu nome e número USP. Não é permitida consulta a livros, internet e outros materiais e pessoas.

1-) Quantas variáveis de identificador A existem no código abaixo (1.0pt)? Escreva literalmente o que o código imprime (0.5pt). Justifique em termos de escopo de variáveis (1.0pt).

```
class Var {
    static int A=10;
    static void f (int A) {
        A=7;
    }
    public static void main (String[] args){
        int A=15;
        f(A);
        System.out.println (A);
    }
}
```

@ - Alguém sabe o que significa declarar uma variável com static?

@ - Não seriam 3 variáveis? Pois dentro do método f, é declarada mais uma variável int de A

@ - @ é mesmo hahah são 3 variáveis A, porque tem três declarações. Esse código vai imprimir 15, pq quando chama o método f (passando A=15, que está dentro do bloco do main), o método vai mudar a variável A= 10 (que está declarada na classe). então quando o main pedir para printar A (é o A que está dentro do main) que não mudou, então 15.

@ - não entendi porque o método não aplica o valor 7 a A=15, se ele vem dentro do main após A=15 -

não é pq ele printa A, e não f(A)? e A é declarado como 15 dentro do main

@ - Mas f(A) é um método que aplica valor 7 a variável A, é como se o código que está dentro do método f fosse aplicado logo após A=15.

@- Pode ser porque o método main é "void" e não retorna nada. É só pegadinha do Nakano. Mas pode existir 2 variáveis com o mesmo nome? Mesmo sendo uma dentro da classe e outra nos métodos?

@- pelo que eu entendi, o método f tá mudando a variável da classe, pq se não ele ele imprimiria 7 mesmo.

@ - mas uma variável declarada dentro da classe passa a ser uma variável global n é?

@- Sim, só que mesmo sendo global acredito que seja possível alterar o valor dela nos métodos. O que me preocupa é aquele "static", não sei qual a função dele.

@- Acho que tem a ver com os escopos de variáveis que ele fala, porque o f é uma variável de instância, declarada dentro de uma classe mas fora do método (main), enquanto o A é a variável estática, que não é modificada por ter essa característica... mas n tenho certeza

@monitor

galera, a variável que seria alterada é a do main, porq? Pq ela é a de escopo mais próximo da chamada do método. Portanto, ela que é enviada. Mas, ela ainda assim não é modificada, pois, além do método ser void, a variável de dentro do método é apenas uma cópia, e não substitui a original que está dentro do main.

@ - Chegou nosso salvador! \o/

@monitor

hahahaha, não esquentar com o static. Aí ele não muda nada. Só quer dizer que a variável GLOBAL A é associada à classe.

E, sim, são 3 variáveis mesmo.

1 global, 1 cópia no método, e 1 no main.

2-) O código abaixo compila, mas quando executado imprime:

```
nakano@nakano:~/Desktop/ach2001-p1$ javac Vet.java
nakano@nakano:~/Desktop/ach2001-p1$ java Vet
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 7 at Vet.f2(Vet.java:5) at
Vet.main(Vet.java:15)
```

a-) Qual é o problema (0.5pt)? b-) Como você o resolveria (1.0pt)? c-) Uma solução seria comentar a invocação (chamada) a f2(...) (esta solução e assemelhadas não são respostas válidas ao item b). Caso isso seja feito, o programa compila e executa. Escreva literalmente o que ele imprime (1.0pt).

```
class Vet {
    static void f2 (int[] A) {
        int i;
        for (i=A.length;i>=0;i--) {
            A[i]=55; // @ linha 4
        }
    }
    static void f(int[] A) {
        for (int i=0;i<A.length;i++) {
            A[i]= A[i]-2*i;
        }
    }
    public static void main (String[] args) {
        int[] A={1,2,3,5,7,11,13}; // @ linha 13
        f2(A);
        f(A);
        for (int i=0;i<A.length;i++) {
            System.out.println (A[i]);
        }
    }
}
```

@ - A) Erro de execução na linha 4 e 13, a variavel "i" recebe um indice do array maior do que existe, exemplo A[7] não existe, mas o i recebe 7 = A.length

@ - B) for (i=0; i<A.length;i++) ou (i=A.length-1;i>=0;i--)

@ - C) Imprime 1 , 0 , -1 , -1 , -1 , 1 , 1

@ - o que é comentar a chamada d ef2?

@ - Seria //f(2), você estaria comentando a invocação do metodo, o programa não iria "ler" e ai não executaria o método dentro do main

+1, valeu

@ - (alternativa c)) Se comentar f2, ele daria erro no main quando tenta executar f2(A) né? E se comentarmos essa parte também, ele somente imprimirá o f(A)...

@ - Ele disse para comentar a invocação do f2. Ou seja, dentro do main. Isso quer dizer que o f2 não seria usado e executaria apenas o método f. Ficaria assim:

```
public static void main (String[] args) {  
    int[] A={1,2,3,5,7,11,13}; // @ linha 13  
    //f2(A);  
    f(A);  
    for (int i=0;i<A.length;i++) {  
        System.out.println (A[i]);  
    }  
}
```

@ - Ahh entendi, ele só comenta dentro do main, achei q era toda a parte do método f2... Vlw Fabiano!!!

@ - não entendi o erro na linha 4, você quis dizer que quando coloca i=A.length quer dizer que i recebe um valor acima da capacidade do array ?

3-) Escreva sucintamente o que você entende sobre os processos de compilação e execução dos programas escritos em **Java. (o que é compilar, o que isso gera, o que é executar, quais são os programas envolvidos,...) (1.0pt)**

@ - O código fonte que é escrito pelo programador é compilado em um arquivo bytecode por meio de um compilador, como o javac. Este bytecode é o que vai ser “transcrito” para o sistema operacional em bytes, e ocorre por meio da máquina virtual Java (JVM). Após essa tradução, o arquivo vira um código em bytes que é executável pelo sistema operacional e pode ser utilizado

@ - Um programa é compilado ao transformar o código fonte em código de máquina, sendo eles respectivamente o arquivo .java e o arquivo .class. O compilador do pacote JDK é o javac, e o programa java é o que executa o bytecode através da JVM (Java Virtual Machine - Máquina Virtual Java).

4-) No código abaixo, a matriz T representa a tela de um computador - ela } tem 1080 linhas e 1920 colunas. Cada elemento representa um ponto de luz. Se o elemento valer zero, então a cor é preta, caso contrário a cor é branca.

Você deve fazer o gráfico de f(x) na tela.

Para isto, cada COLUNA corresponde a um valor de x - por exemplo para x=0 a coluna é 0, para x=1 a coluna é 1, para x=2, a coluna é 2 e assim por diante. Cada LINHA corresponde a um valor de y - por exemplo, para y=0 a linha é 0, para y=1 a linha é 1, para y=2 a linha é 2 e assim por diante.

Para cada valor x , existe um valor $y=f(x)$. O par (x,y) corresponde a um ponto que precisa receber a cor branca.

a-) Escreva o método `desenha()`, que usa o método `f(x)` e para cada x no intervalo $[0..1919]$ calcula o valor de y correspondente e "colore" o respectivo ponto (x,y) com "branco" (1.5pt). Tenha o cuidado de não "desenhar fora da Tela" (1.0pt).

@ Alguém manjou essa?

@ - tentativa:

```
static void desenha () {  
  
    int y=0;  
  
    for(int x=0;x<T.length;x++){  
  
        y= f(x);  
  
        if (y>=0 && y<T[0].length){  
  
            T[x][y]=1;  
  
            System.out.println("As coordenadas (" +x+" "+y+" ) foram pintadas");  
  
        }  
  
    }  
  
}
```

@ - Alguém confirma se está certo/errado?

@ - Do jeito que vc fez imprime só isso:
"As coordenadas (0,2) foram pintadas"

As coordenadas (1,0) foram pintadas

As coordenadas (2,2) foram pintadas"

mas se colocar " $y= -f(x)$;" ele imprime tudo eu acho.

@i - é que tirando essas 3 coordenadas, segundo o código "`return x^2 -x`" sempre retornará resultados negativos, e y não pode ser negativo

@ - o que exatamente o "x^2-x" faz? Não entendi a lógica dos resultados que aparecem.

@ - Bela pergunta HAUSAHS, em outras linguagens de programação, seria potencia, mas não em java

```
class Tela { static int[][] T; static int f(int x) { return x^2-x; }  
static void desenha () {
```

@ - Update: essa foi a última tentativa, vou dormir senão aí vou mal na prova!! O código imprime vários 1s e 0s alternadamente de modo infinito...

```
class Tela {  
    static int[][] T;  
  
    static int f(int x) {  
        return (x^2)-x;  
    }  
  
    static void desenha (int[][] A) {  
        int w = 0; // Aqui y = linha e x = coluna  
        for (int x=0; x < 1920; x++) { // Não consegui usar length para obter o nº 1920  
            w = f(x);  
            for (int y=0; y < A.length; y++){  
                if (w>=0 && w < A[y].length) {  
                    // A[y][x] = 1;  
                    System.out.print ("1 ");  
                }  
                else System.out.print ("0 ");  
            }  
        }  
    }  
  
    public static void main (String[] args) {  
        T=new int[1080][1920]; // T é uma tela preta  
        desenha(T);  
    }  
}
```

@ - Pessoal, no 4 o 0 representa preto e o 1 representa branco?

Sim

/* seu código deve vir aqui */

```
public static void main (String[] args) {
    T=new int[1080][1920]; // T é uma tela preta
    desenha();
}
```

(0,0)									

5-) Complete o quadro com os números nas bases indicadas (1.5pt).

binario	octal	decimal	hexadecimal
			D10
		91	
	225		

}

@ D10 - Binário :

“D” corresponde a 13ª combinação de binários, que vai de “0” - 0000 até “F” - 1111.
Desta forma, “D” é 1101.

“1” corresponde a 0001 (jura?).

“0” corresponde a 0000.

O número em binário fica como 110100010000.

D10 - Octal

Pra passar pra octal, é mais fácil utilizar o número em binário. O código hexadecimal lê o binário em blocos de 4 dígitos, no caso "1101", "0001" e "0000". Para octal, em blocos de 3, logo "110", "100", "010", "000". Convertendo estes blocos pra octal, que vai de "0" - 000 até "7" - 111, ficaria 110 -> "6", 100 -> "4", 010 -> "2" e 000 -> "0", ou seja, **6420**.

D10 - Decimal

Pra passar pra decimal também é mais fácil utilizar o número em binário. Pega a cadeia

110100010000 e soma as potências de 2 nos dígitos com 1. No caso, vai até 2^{11}
 $1 \cdot 2^{11} + 1 \cdot 2^{10} + 0 \cdot 2^9 + 1 \cdot 2^8 + \dots + 1 \cdot 2^4 + \dots$, o que resulta em $2048 + 1024 + 256 + 16$, que é igual a 3344... @ - acho que da 3344 - @Lucas B corrigido, escrevi errado mesmo

Para realizar as outras conversões é só seguir os mesmos passos, converte sempre pra binário e usa ele como base que fica muito mais fácil pras outras bases

D10 não seria D e 10?

-> não porque você lê bloco por bloco. E o "10" em hexadecimal é dado como A. então se fosse 10 mesmo o hexadecimal seria DA

@ Outra opção para converter Hexadecimal em Decimal é multiplicar os dígitos por 16 (valor da base hexadecimal) elevados às potências de 0 a N e depois somá-los, conforme mostra abaixo:

$$D_{(16)} + 1_{(16)} + 0_{(16)} = 13_{(10)} + 1_{(10)} + 0_{(10)}$$

$$(13 \cdot 16^2) + (1 \cdot 16^1) + (0 \cdot 16^0) = 3328 + 16 + 0 = 3344$$

$$\text{Portanto } D10_{(16)} = 3344_{(10)}$$

@

Decimal – 91

Binário:

$$91/2 = 45 \rightarrow \text{Resto1} = 1$$

$$45/2 = 22 \rightarrow \text{Resto2} = 1$$

$$22/2 = 11 \rightarrow \text{Resto3} = 0$$

$$11/2 = 5 \rightarrow \text{Resto4} = 1$$

$$5/2 = 2 \rightarrow \text{Resto5} = 1$$

$$2/2 = 1 \rightarrow \text{Resto6} = 0$$

$$1/2 = 0 \rightarrow \text{Resto7} = 1$$

$$\text{Resto7} + \text{Resto6} + \text{Resto5} + \text{Resto4} + \text{Resto3} + \text{Resto2} + \text{Resto1} =$$

$$1 + 0 + 1 + 1 + 0 + 1 + 1$$

$$\text{Portanto } 91_{(10)} = 1011011_{(2)}$$

$$\text{Octal} - 001_{(2)} + 011_{(2)} + 011_{(2)} = 1_{(8)} + 3_{(8)} + 3_{(8)} \text{ Portanto } 1010011_{(2)} = 133_{(8)}$$

$$\text{Hexadecimal} - 0101_{(2)} + 1011_{(2)} = 5_{(16)} + B_{(16)} \text{ Portanto } 1010011_{(2)} = 5B_{(16)}$$

Octal - 225

$$\text{Binário} - 2_{(8)} + 2_{(8)} + 5_{(8)} = 010_{(2)} + 010_{(2)} + 101_{(2)} \text{ Portanto } 225_{(8)} = 10010101_{(2)}$$

Decimal:

Um método prático de converter binário para decimal é alinhar as potências de 2 com os dígitos do número binário e somar apenas os valores que estejam alinhados com 1 (ignorar os valores alinhados com 0)

$$\begin{array}{cccccccc} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

$$2^7 + 2^4 + 2^2 + 2^0 = 128 + 16 + 4 + 1 = 149$$

$$\text{Portanto } 10010101_{(2)} = 149_{(10)}$$

$$\text{Hexadecimal} = 1001_{(2)} + 0101_{(2)} = 9_{(16)} + 5_{(16)} \text{ Portanto } 10010101_{(2)} = 95_{(16)}$$

Decimal	Binário	Octal	Hexadecimal
00	0000	00	0
01	0001	01	1
02	0010	02	2

03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F