Unidade I:

Conceitos Básicos - Exercícios Iniciais

Prof. Max do Val Machado

Instituto de Ciências Exatas e Informática
Curso de Ciência da Computação

Potência de Dois

- Para a aula de análise de complexidade, resolva as equações abaixo:

a) $2^0 =$

b) $2^1 =$

c) $2^2 =$

d) $2^3 =$

e) $2^4 =$

f) $2^5 =$

g) $2^6 =$

h) $2^7 =$

i) $2^8 =$

j) $2^9 =$

k) $2^{10} =$

l) $2^{11} =$

Logaritmo na Base Dois

- Para a aula de análise de complexidade, resolva as equações abaixo:

a) $\lg(2048) =$
b) $\lg(1024) =$
c) $\lg(512) =$
d) $\lg(256) =$
e) $\lg(128) =$
f) $\lg(64) =$
g) $\lg(32) =$
h) $\lg(16) =$
i) $\lg(8) =$
j) $\lg(4) =$
k) $\lg(2) =$
l) $\lg(1) =$

Piso e Teto

- Para a aula de análise de complexidade, resolva as equações abaixo:

a) $\lceil 4{,}01 \rceil =$

b) $\lfloor 4{,}01 \rfloor =$

c) $\lceil 4{,}99 \rceil =$

d) $\lfloor 4{,}99 \rfloor =$

e) $\lg(16) =$

f) $\lceil \lg(16) \rceil =$

g) $\lfloor \lg(16) \rfloor =$

h) $\lg(17) =$

i) $\lceil \lg(17) \rceil =$

j) $\lfloor \lg(17) \rfloor =$

k) $\lg(15) =$

l) $\lceil \lg(15) \rceil =$

m) $\lfloor \lg(15) \rfloor =$

Gráfico de Funções

- Plote um gráfico com todas as funções abaixo:

a) $f(n) = n$

b) $f(n) = n^2$

c) $f(n) = n^3$

d) $f(n) = sqrt(n)$

e) $f(n) = lg\ (n)$

f) $f(n) = 3n^2 + 5n - 3$

g) $f(n) = - 3n^2 + 5n – 3$

h) $f(n) = | - n^2 |$

i) $f(n) = 5n^4 + 2n^2$

j) $f(n) = n * lg\ (n)$

Exercício

- Faça um método que receba um *array* de inteiros e um número inteiro *x* e retorne um valor booleano informando se o elemento *x* está contido no *array*

- Repita o exercício acima considerando que os elementos do *array* estão ordenados de forma crescente. Uma sugestão seria começar a pesquisa pelo meio do *array*

Exercício

- Faça um método que receba um *array* de inteiros e mostre na tela o maior e o menor elementos do *array*.

- Repita o exercício acima fazendo menos comparações com os elementos do *array*

Exercício

- O que o código abaixo faz?

```
boolean doidao (char n){
        boolean resp= false;
        int v = (int) c;
        if (v == 65 || v == 69 || v == 73 || v == 79 || v == 85 || v == 97 || v == 101 || v ==105 ||
            v == 111 || v == 117){
              resp = true;
        }
        return resp;
}
```

Exercício

- O que o código abaixo faz?

```
boolean doidao (char n){
      boolean resp= false;
      int v = (int) c;
      if (v == 65 || v == 69 || v == 73 || v == 79 || v == 85 || v == 97 || v == 101 || v ==105 ||
          v == 111 || v == 117){
            resp = true;
      }
      return resp;
}

char toUpper(char c){
      return (c >= ‘a' && c <=  ‘z') ? ((char) (c – 32)) : c ;
}

boolean isVogal (char c){
      c = toUpper(c);
      return (c =='A' || c =='E' ||  c =='I' ||  c =='O' || c =='U');
}
```

Fazendo o isVogal, fica mais fácil ...

Exercício

- Outras opções

```
boolean isLetra (char c){
	return (c >= 'A' && c <='Z' ||  c >='a' &&  c <='Z');
}

boolean isConsoante (char c){
	return (isLetra(c) == true &&  isVogal(c) == false);
}
```

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int n){
    boolean resp= true;
    if (n!=s.length()){
      if (s.charAt(n)<'0' || s.charAt(n)>'9'){
        if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
          s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
          s.charAt(n)=='o' || s.charAt(n)=='u'){
          resp= false;
        } else{
          n++;
          resp=isConsoante(s, n);
        }
      } else {
        resp=false;
      }
    }
    return resp;
  }
```

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int n){
    boolean resp= true;
    if (n!=s.length()){
      if (s.charAt(n)<'0' || s.charAt(n)>'9'){
        if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
          s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
          s.charAt(n)=='o' || s.charAt(n)=='u'){
          resp= false;
        } else{
          n++;
          resp=isConsoante(s, n);
        }
      } else {
        resp=false;
      }
    }
    return resp;
  }
```

1º passo: Indentação

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int n){
	boolean resp= true;
	if (n != s.length()){
		if (s.charAt(n)<'0' || s.charAt(n)>'9'){
			if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
			  s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
			  s.charAt(n)=='o' || s.charAt(n)=='u'){
				resp= false;
			} else {
				n++;
				resp=isConsoante(s, n);
			}
		} else {
			resp=false;
		}
	}
	return resp;
}
```

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int n){
        boolean resp= true;
        if (n != s.length()){
                if (s.charAt(n)<'0' || s.charAt(n)>'9'){
                        if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
                          s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
                          s.charAt(n)=='o' || s.charAt(n)=='u'){
                                resp= false;
                        } else {
                                n++;
                                resp=isConsoante(s, n);
                        }
                } else {
                        resp=false;
                }
        }
        return resp;
}
```

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int n){
      boolean resp= true;
      if (n != s.length()){
            if (s.charAt(n)<'0' || s.charAt(n)>'9'){
                  if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
                    s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
                    s.charAt(n)=='o' || s.charAt(n)=='u'){
                        resp= false;
                  } else {
                        n++;
                        resp=isConsoante(s, n);
                  }
            } else {
                  resp=false;
            }
      }
      return resp;
}
```

2º passo: Simplificação

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int n){
      boolean resp= true;
      if (n != s.length()){
            if (s.charAt(n)<'0' || s.charAt(n)>'9'){
                  if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
                     s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
                     s.charAt(n)=='o' || s.charAt(n)=='u'){
                        resp= false;
                  } else {
                        n++;
                        resp=isConsoante(s, n);
                  }
            } else {
                  resp=false;
            }
      }
      return resp;
}
```

2º passo: Simplificação

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int n){
	boolean resp= true;
	if (n != s.length()){
		if (s.charAt(n)<'0' || s.charAt(n)>'9'){
			if (isVogal(s.charAt(n))  ==  true){
				resp= false;
			} else {
				resp=isConsoante(s, n + 1);
			}
		} else {
			resp=false;
		}
	}
	return resp;
}
```

2º passo: Simplificação

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int n){
	boolean resp= true;
	if (n != s.length()){
		if (s.charAt(n)<'0' || s.charAt(n)>'9'){
			if (isVogal(s.charAt(n))  ==  true){
				resp= false;
			} else {
				resp=isConsoante(s, n + 1);
			}
		} else {
			resp=false;
		}
	}
	return resp;
}
```

O código está correto?

Exercício Resolvido

- Um aluno desenvolveu o código abaixo, corrija-o:

```
boolean isConsoante(String s, int i){
	boolean resp= true;

	if (i == s.length()){
		resp = true;
	} else if (isConsoante(s.charAt(i)) == false){
		resp = false;
	} else {
		resp = isConsoante(s, i + 1);
	}

	return resp;
}
```

Exercício

- Qual das duas versões é mais fácil de entender?

```
boolean isConsoante(String s, int i){
        boolean resp= true;

        if (i == s.length()){
                resp = true;
        } else if (isConsoante(s.charAt(i)) == false){
                resp = false;
        } else {
                resp = isConsoante(s, i + 1);
        }

        return resp;
}
```

```
boolean isConsoante(String s, int i){
        boolean resp= true;

        if (i < s.length()){
                if (!isConsoante(s.charAt(i))){
                        resp = false;
                } else {
                        resp = isConsoante(s, i + 1);
                }
        } else {
                resp = true;
        }

        return resp;
}
```

Exercício

- Qual é a sua opinião sobre o código **REAL** abaixo?

```
Unidade recuperarUnidadeComCodigoDeUCI(Unidade unidadeFilha) {
	Unidade retorno = null;

	if (unidadeFilha.getCodUci() != null && !unidadeFilha.getCodUci().isEmpty()) {
		retorno = unidadeFilha;
	} else {
		retorno = unidadeFilha.getUnidadeSuperior();
	}

	while (retorno == null || retorno.getCodUci() == null || retorno.getCodUci().isEmpty()) {
		retorno = retorno.getUnidadeSuperior();
	}

	return retorno;
}
```

Exercício

- Qual é a diferença entre os dois métodos abaixo?

```
int m1(int i){
        return i--;
}


int m2(int i){
        return --i;
}
```

Exercício

- O que o programa abaixo mostra na tela?

```
byte b = 0; short s = 0; int i = 0; long l = 0;

while (true){
      b++;    s++;    i++;    l++;
      System.out.println(b + " " + s + " " + i + " " + l);
}
```

Exercício Resolvido

- Por que o código abaixo imprime [46 - 11]?

```
int x = 23, y = 23;
x = x << 1;
y = y >> 1;
System.out.println("[" + x + " - " + y + "]");
```

Exercício Resolvido

- Por que o código abaixo imprime [46 - 11]?

```
int x = 23, y = 23;
x = x << 1;
y = y >> 1;
System.out.println("[" + x + " - " + y + "]");
```

Os operadores *shift right* e *left* (>> e <<) deslocam os bits para direita e esquerda e inserem um zero na posição vazia

Na prática, temos, uma divisão ou multiplicação por dois

Exercício Resolvido

- Por que o código abaixo imprime [46 - 11]?

```
int x = 23, y = 23;
x = x << 1;
y = y >> 1;
System.out.println("[" + x + " - " + y + "]");
```