

Algoritmos e Estruturas de Dados I (DCC/003)

Lista de Exercícios 05 – Revisão de Conceitos

Nome: Guilherme de Abreu Lima Buitrago Miranda

Matrícula: 2018054788

Exercício 1

- a) Um programa de computador é um software programado em alguma linguagem de programação e que pode ou não passar por um compilador. Com isso, o programa executa, junto do computador, baseado na interação com o usuário, uma série de instruções implementadas pelo desenvolvedor, a fim de realizar determinada tarefa.
- b) Um compilador é uma ferramenta utilizada na maior parte das linguagens de programação para transformar o código escrito pelo desenvolvedor em uma linguagem de mais baixo nível, a ser executada pelo processador do computador.
- c) Um erro de sintaxe pode ser um comando escrito de maneira a não ser reconhecido pelo compilador, o que impede o programa de executar. Alguns exemplos são a falta de ponto e vírgula em um comando ou a falta de chaves em uma estrutura de repetição.
- d) Um erro de lógica é aquele que impede com que o programa cumpra a função para a qual ele foi designado, ou seja, o programa é aceito pelo compilador e executado, mas não gera um resultado correto. Um exemplo de erro de lógica seria uma comparação de menor ou igual, quando, na verdade, o desenvolvedor deveria comparar apenas se o valor era menor.
- e) Um algoritmo é uma sequência de passos a fim de cumprir determinada tarefa. Os algoritmos são bastante utilizados na área de computação, pois é dessa maneira que programas são escritos para a execução em computadores.
- f) Uma variável é um local na memória onde são alocados valores. Um exemplo de variável é a idade do usuário, que armazena um número inteiro e pode ser utilizado como dado para a execução de um programa.
- g) Os tipos de dados são as diferentes formas que um tipo pode ser representado numa linguagem. Alguns dos tipos mais famosos são os inteiros, os reais, os caracteres, as strings e o tipo booleano (verdadeiro ou falso). Além disso, os tipos podem variar de acordo com a precisão e a utilização de memória, como os tipos float e double.

- h) A notação binária é uma das maneiras de se representar um número qualquer. Na computação a notação binária é bastante utilizada pois é a forma com que o computador trabalha no mais baixo nível, representando números apenas com 0's e 1's. Um exemplo de notação binária é o número 1001, que, no sistema decimal, representa o número 9.
- i) A notação hexadecimal é uma outra maneira de se representar um número qualquer. Ela varia do número 0 ao 9 e do caractere a ao f, e também é bastante utilizada na computação, sobretudo na representação de posições de memória. Um exemplo de notação hexadecimal é o número 38f, que representa o número 911 em decimal.

2) Na linguagem C existem 3 tipos principais de variáveis: char, int e float. Além disso, o char pode não ser assinado, o que é representado por "unsigned char". O int pode ser short (curto), unsigned (não assinado), além de unsigned short. O int também pode ser long, assim como unsigned long. Por fim, o float pode ser double e long double, variando da precisão nas casas flutuantes.

3) Os operadores aritméticos da linguagem C são: +, -, *, / e %, e representam, respectivamente: soma, subtração, multiplicação, divisão e resto da divisão inteira.

4) Os seis operadores relacionais da linguagem C são: ==, !=, >=, <=, > e <, e representam, respectivamente, igual, diferente, maior ou igual, menor ou igual, maior e, por fim, menor.

5) O scanf é o comando utilizado para realizar a leitura de variáveis fornecidas pelo usuário. Um exemplo em que o usuário passa dois valores a serem lidos pode ser representado por:

```
scanf("%d %d", &numeroA, &numeroB);
```

6) O comando printf é utilizado para representar saídas do programa. Nele, podem ser impressas mensagens para o usuário, assim como valores de variáveis calculadas pelo programa. Um exemplo em que o programa mostra ao usuário um valor inteiro e um valor fracionário pode ser representado por:

```
printf("A parte inteira do número = %d e a parte fracionária = %f", parteInteira, parteFrac);
```

7) A estrutura condicional if-else é utilizada para controle de execução do programa a ser desenvolvido. No caso, se a condição do if for verdadeira, uma sequência de comando é executada e, caso contrário, o bloco de comandos do else é executado. Um exemplo de comando if-else para fazer uma ou outra saída para o usuário pode ser representada por:

```
if (primo == 0){  
    printf("O numero %d é primo\n", numero);  
}else{
```

```
    printf("O numero %d não é primo\n", numero);  
}
```

8) Uma função em linguagem C serve para executar uma sequência de comandos e retornar, ou não, um valor para a função principal (main). É bastante utilizada para evitar a repetição de código, pois pode ser chamada inúmeras vezes sem a necessidade de reescrita. Ela deve ser definida com o tipo de retorno e as variáveis necessárias para sua execução.

```
9) int somaNumeros (int a, int b){  
    return (a+b);  
}
```

10) O comando while serve para repetir um bloco de código até que a condição dada seja falsa. Um excelente exemplo desse comando é um algoritmo que calcule o resto e a parte inteira de uma divisão, como representado no exemplo a seguir:

```
int quoc = 0;  
while(num1 >= num2){  
    quoc += 1;  
    num1 -= num2;  
}  
  
int resto = num1;  
printf("O resto da divisão é igual a: %d e a parte inteira igual a %d \n", resto, quoc);
```

Nesse caso, o while se repete até que o número 1 seja maior ou igual ao número 2.

11) O comando for serve para executar um bloco de código por repetidas vezes até que um número (comumente chamado de contador) atinja outro parâmetro. Um exemplo para compreender a dinâmica de funcionamento do for pode ser representada no seguinte exemplo, que representa a sequência de Fibonacci até um número n qualquer.

```
int a = 0;  
int b = 1;  
int aux;  
for (int i = 0; i < numero; ++i){  
    printf("%d ", b);  
    aux = b;  
    b = a + b;
```

```
    a = aux;  
}
```

Nesse caso, ao final do for, o número n da sequência será representado pela variável b.

Sugestão: Ok. Tudo ficou extremamente claro ao decorrer das aulas.