## Fatec - Algoritmos e Programação de Computadores

## Lista de Exercícios 2

- Faça um programa que imprima um menu de 4 pratos na tela e uma quinta opção para sair do programa. O programa deve imprimir o prato solicitado. O programa deve terminar quando for escolhido a quinta opção.
- 2. Faça um programa que lê dois números inteiros positivos a e b. Utilizando laços, o seu programa deve calcular e imprimir o valor  $a^b$ .
- 3. Faça um programa que lê um número n e que compute e imprima o valor

$$\sum_{i=1}^{n} i.$$

OBS: Não use formulas como a da soma de uma P.A.

- 4. No exemplo dos números primos visto em aula, não precisamos testar todos os números entre  $2, \ldots, (n-1)$ , para verificar se dividem ou não n. Basta testarmos até n/2. Por que? Qual o maior divisor possível de n? Na verdade basta testarmos os números  $2, \ldots, \sqrt{n}$ . Por que?
- 5. Considere o programa para determinar se uma sequência de n números digitados pelo usuário está ordenada ou não. Faça o programa usando uma variável contadora.
- 6. Faça um programa em C que calcule o máximo divisor comum de dois números m, n. Você deve utilizar a seguinte regra do cálculo do mdc com  $m \ge n$

$$mdc(m,n) = m \text{ se } n = 0$$
  
 $mdc(m,n) = mdc(n,m\%n) \text{ se } n > 0$ 

- 7. Escreva um programa que lê um número n, e então imprime o menor número primo que é maior ou igual n, e imprime o maior primo que é menor ou igual a n.
- 8. O que será impresso pelo programa abaixo? Assuma que o valor de D na declaração de  $\mathbf{x}$  é o valor do último dígito do seu RA.

```
int main() {
  int x = 5+D, y = 0;
  do {
    y = (x % 2) + 10 * y;
    x = x / 2;
    printf("x = %d, y = %d\n", x, y );
} while (x != 0);
while (y != 0) {
```

```
x = y % 100;
y = y / 10;
printf("x = %d, y = %d\n", x, y );
}
}
```

9. Escreva um programa para ler n de números do tipo float e imprimir quantos deles estão nos seguintes intervalos:  $[0\dots25], [26\dots50], [51\dots75]$  e  $[76\dots100]$ . Por exemplo, para n=10 e os seguintes dez números 2.0, 61.5, -1.0, 0.0, 88.7, 94.5, 55.0, 3.1415, 25.5, 75.0, seu programa deve imprimir:

Intervalo [0..25]: 3
Intervalo [26..50]: 0
Intervalo [51..75]: 3
Intervalo [76..100]: 2

10. Escreva um programa em C para computar a raiz quadrada de um número positivo. Use a idéia abaixo, baseada no método de aproximações sucessivas de Newton. O programa deverá imprimir o valor da vigésima aproximação.

Seja Y um número, sua raiz quadrada é raiz da equação

$$f(x) = x^2 - Y.$$

A primeira aproximação é  $x_1 = Y/2$ . A (n+1)-ésima aproximação é

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

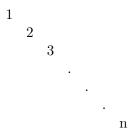
11. Aponte os erros de implementação existentes no código em C, a seguir, desenvolvido com o intuito de calcular e imprimir o *fatorial* de um número inteiro não-negativo.

12. Implemente um programa que compute todas as soluções de equações do tipo

$$x_1 + x_2 + x_3 + x_4 = C$$

onde todas as variáveis  $x_1, \ldots, x_4$  são inteiras não negativas e C > 0 é uma constante inteira. Melhore o seu programa com as seguinte idéias.

- Fixado  $x_1$ , os valores possíveis para  $x_2$  são  $0, \ldots, C x_1$ . Fixado  $x_1$  e  $x_2$ , os valores possíveis para  $x_3$  são  $0, \ldots, C x_1 x_2$ . Fixados  $x_1, x_2$ , e  $x_3$ , então  $x_4$  é unicamente determinado.
- 13. Na transformação decimal para binário, modifique o programa para que este obtenha o valor binário em uma variável inteira, ao invés de imprimir os dígitos um por linha na tela. **Dica:** Suponha n=7 (111 em binário), e você já computou x=11, para "inserir"o último dígito 1 em x você deve fazer x=x+100. Ou seja, você precisa de uma variável acumuladora que armazena as potências de 10: 1, 10, 100, 1000 etc.
- 14. Faça um programa que leia um inteiro n (no máximo 50) e imprima uma saída da forma:



15. Faça um programa que leia um número n e imprima n linhas na tela com o seguinte formato (exemplo se n=6):

```
1 2 1 2 3 1 2 3 4 1 2 3 4 5 5 1 2 3 4 5 6
```

16. Faça um programa que leia um número n e imprima n linhas na tela com o seguinte formato (exemplo se n=6):

17. Um jogador da Mega-Sena é supersticioso, e só faz jogos em que o primeiro número do jogo é par, o segundo é ímpar, o terceiro é par, o quarto é ímpar, o quinto é par e o sexto é ímpar. Faça um programa que imprima todas as possibilidades de jogos que este jogador supersticioso pode jogar.