

EX - Lista de exercicios de funções

Prof.: Paulo Roberto Nunes de Souza

1 Exercícios sobre funções

1.1. Utilizando o código chamado *retangulo.c* mostrado a seguir.

Arquivo *retangulo.c*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 float areaRetangulo(float b, float h){
5
6 }
7
8 int main(){
9     float a, b, r;
10    printf("Digite o tamanho da base do retangulo: ");
11    scanf("%f", &a);
12
13    printf("Digite a altura do retangulo: ");
14    scanf("%f", &b);
15
16    r = areaRetangulo(a, b);
17
18    printf("A area do retangulo eh %f\n", r);
19    return 0;
20 }
```

Termine de escrever a função *areaRetangulo*. Com a função corretamente terminada, o programa em linguagem C deverá receber dois valores reais que representem a base e a altura de um retângulo. A seguir deve utilizar a função *areaRetangulo* para calcular a área do retângulo e ao final imprimir na tela o valor calculado da área deste retângulo.

1.2. Utilizando o código chamado *maior.c* mostrado a seguir.

Arquivo maior.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void imprimeMaior(int a, int b){
5
6 }
7
8 int main(){
9     int x, y;
10
11     printf("Digite um numero inteiro: ");
12     scanf("%d", &x);
13
14     printf("Digite outro numero inteiro: ");
15     scanf("%d", &y);
16
17     imprimeMaior(x, y);
18
19     return 0;
20 }
```

Termine de escrever a função *imprimeMaior*. Com a função corretamente terminada, o programa em linguagem C deverá receber dois valores inteiros e passá-los como parâmetro para a função *imprimeMaior*. A função deve então avaliar qual dos dois números é o maior e imprimir na tela **apenas** o maior deles.

1.3. Utilizando o código chamado *crescente.c* mostrado a seguir.

Arquivo crescente.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define N 5
5
6 void crescDesc(int a, int b){
7
8 }
9
10 int main(){
11     int i, vet[N];
12
13     printf("Digite %d numeros inteiros\n", N);
14
15     for(i = 0; i < N; i++)
16         scanf("%d", &vet[i]);
17
18     for(i = 0; i < N-1; i++)
19         crescDesc(vet[i], vet[i+1]);
20
21     return 0;
22 }
```

Termine de escrever a função *crescDesc*. Com a função corretamente terminada, o programa em linguagem C deverá receber cinco valores inteiros e armazená-los em um vetor de cinco posições. A seguir o programa deve percorrer o vetor e passar para a função *crescDesc* todos os pares de elementos sucesivos do vetor. Para cada par de valores recebidos, a função deve avaliar os dois valores e imprimir na tela se os valores estão em ordem crescente ou decrescente.

1.4. Utilizando o código chamado *rendaAnual.c* mostrado a seguir.Arquivo *rendaAnual.c*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define N 5
5
6 void imprimeRendaAnual(float salario){
7     printf("Salario Anual e %f\n", 12 * salario);
8 }
9
10 void aumentaSalario(float *salario, float aumento){
11 }
12
13
14 int main(){
15     int i, k;
16     float aumento, sal[N];
17
18     printf("Digite %d numeros reais (salario mensal)\n", N);
19
20     for(i = 0; i < N; i++)
21         scanf("%f", &sal[i]);
22
23     printf("Digite um numero inteiro de 0 a %d\n", N - 1);
24
25     scanf("%d", &k);
26
27     printf("Digite um numero real que represente quanto deve ser dado de aumento\n");
28     scanf("%f", &aumento);
29
30     aumentaSalario(&sal[k], aumento);
31
32     for(i = 0; i < N; i++)
33         imprimeRendaAnual(sal[i]);
34
35     return 0;
36 }
```

Termine de escrever a função *aumentaSalario*. Com a função corretamente terminada, o programa em linguagem C deverá receber cinco valores reais e armazená-los em um vetor de cinco posições. Cada valor real lido representa o salário mensal de um funcionário. Em seguida o programa deverá receber um número inteiro entre 0 e 4 que represente um dos funcionários cujo salário foi armazenado no vetor. Por último, o programa deve receber um valor real que indique qual é o valor do aumento que o funcionário indicado anteriormente deve receber. Após receber todas as informações o programa deve acionar a função *aumentaSalario* para que ela altere o salário do funcionário indicado, acrescentando o valor informado. Ao final o programa deverá imprimir a renda anual dos cinco funcionários utilizando a função *imprimeRendaAnual*.

1.5. Utilizando o código chamado *saque.c* mostrado a seguir.

Arquivo saque.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void sacar(float *saldo, float saque){
5
6 }
7
8 int main(){
9     float saldo, saque = 1.0;
10    printf("Digite o saldo inicial (numero real): ");
11    scanf("%f", &saldo);
12
13    while(saque > 0.0){
14        printf("Digite o valor a ser sacado: ");
15        scanf("%f", &saque);
16        sacar(&saldo, saque);
17    }
18
19    return 0;
20 }
```

Termine de escrever a função *sacar*. Com a função corretamente terminada, o programa em linguagem C deverá receber um valor real que represente o saldo de uma conta bancária. A seguir o programa deve receber um valor a ser sacado. De posse do valor, o programa deverá acionar a função *sacar*. A função deve verificar se o valor a ser sacado é menor que o saldo disponível, caso seja, a função deve diminuir o valor sacado do saldo e imprimir na tela uma mensagem informando que o saque foi feito, caso contrário, a função deve informar que o saldo é insuficiente. Caso seja informado um valor de saque menor ou igual a zero, o programa deve se encerrar.