Aula 7 – Sub-rotinas

1. Motivação

- Economizar tempo de programação e espaço (de código);
- Maior clareza aos programas, que ficam mais fáceis de ler e entender;
- Facilitar a manutenção (mudar a sub-rotina em um único lugar).

Utilizadas quando parte do programa é repetida várias vezes. Exemplos:

- Calcular o IMC de uma pessoa;
- Calcular a média de um aluno;
- Calcular o MDC entre dois números;
- Abrir, fechar ou imprimir dados em um arquivo, etc.

2. Tipos de sub-rotinas

- Procedimentos: não retornam valor;
- Funções: retornam um valor como resultado do processamento.

3. Exemplo

```
#include <stdio.h>

float imc;

void imprimirIMC() {
     printf("O seu IMC é: %.1f\n", imc);
}

float calcularIMC(float altura, float peso) {
     return peso / (altura * altura);
}

main() {
     float altura, peso;

     printf("Bom dia! Qual é a sua altura?\n");
     scanf("%f", &altura);
     printf("E o seu peso?\n");
     scanf("%f", &peso);

     imc = calcularIMC(altura, peso);
     imprimirIMC();
}
```

Com auxílio do exemplo, explicar cada característica das sub-rotinas:

- Procedimentos possuem tipo de retorno `void`, funções possuem qualquer outro tipo;
- Em seguida, aparece o nome da sub-rotina;
- Em seguida, entre parênteses, os tipos e nomes dos parâmetros;
- Finalmente, abre-se um bloco (abre-chaves), como na função `main()`. Até o fechamento do bloco fica o código da sub-rotina, que é como um programa: declara-se as variáveis, executa-se comandos e pode chamar outras sub-rotinas;
- Nas funções, para retornar um valor deve-se usar a palavra-chave `return`. Esta palavra-chave pode ser usada também em procedimentos para interromper e sair da sub-rotina;
- Para chamar uma sub-rotina, usa-se: `subRotina(p1, p2, ..., pn)`. Quando função, produz um resultado que pode ser atribuído a uma variável, impresso, etc.
- Parâmetros são passados por valor: alterações em `altura` dentro de `calcularIMC()` não afetam a variável `altura` dentro de `main()`;
- Podemos definir variáveis globais declarando-as fora de qualquer bloco de função: a mesma variável `imc` é acessível dentro de `imprimirIMC()`, `calcularIMC()` e `main()`.

4. Outro exemplo

Fazer com os alunos um programa que calcule a somatória abaixo, dado o número de termos. Construir uma função que retorne o fatorial de um número.

$$e = 1 + \frac{1}{1} + \frac{1}{1\times2} + \frac{1}{1\times2\times3} + \frac{1}{1\times2\times3\times4} + \cdots$$

```
#include <stdio.h>

int calcularFatorial(int n) {
     int fat = 1;
     while (n > 1) {
          fat = fat * n;
          n--;
     }
     return fat;
}

main() {
     float soma = 1;
     int i, n;

     printf("Quantos termos?\n");
     scanf("%d", &n);
     n--;

     for (i = 1; i <= n; i++)
          soma += 1.0 / calcularFatorial(i);
     printf("e = %f\n", soma);
}
```

5. Funções recursivas

Alterar a função fatorial escrita anteriormente para sua forma recursiva.

```
int calcularFatorial(int n) {
     if (n <= 1) return n;
     else return n * calcularFatorial(n - 1);
}
```

6. A função main()

- A `main()` também é uma função, porém especial: ela é chamada pelo sistema operacional quando executamos nosso programa;
- Até agora a utilizamos em sua forma mais simples:

```
main() { }
```

- Porém sua assinatura completa é a seguinte:

```
int main(int argc, char **argv) { }
```

- Seu tipo de retorno serve para avisar o sistema operacional se houve algum erro. Por exemplo, o manual do comando `cp` no Linux indica `0` para sucesso e `> 0` para erros;
- Seus parâmetros servem para passar argumentos da linha de comando para o programa, porém para compreendê-los é preciso primeiro aprender sobre variáveis indexadas e ponteiros;
- Os nomes dos argumentos podem variar.

Universidade Federal do Espírito Santo
Centro Tecnológico
Departamento de Informática

Exercícios – Sub-rotinas

Reescreva os exercícios abaixo (feitos em aulas anteriores) utilizando sub-rotinas onde for apropriado.

Aula 5, exercício 2:

Uma empresa decide dar um reajuste a seus funcionários de acordo com o seguinte critério: 50% de aumento para os que ganham menos de R$ 3.000,00, 20% para os que ganham entre R$ 3.000,00 e R$ 10.000,00 (inclusive) e 15% para os demais. Escreva um algoritmo para calcular este reajuste.

Escreva uma função para calcular o reajuste, dado o salário.

Lista 1, exercício 2:

Faça um programa para ler a altura e a largura de um retângulo e calcular o seu perímetro. Em seguida, faça a mesma coisa, só que recebendo como dados de entrada as coordenadas (x, y) dos cantos inferior esquerdo e superior direito.

Escreva uma função para calcular o perímetro, dados os tamanhos dos lados.

Lista 1, exercício 4:

Considere o valor de π = 3.141592. Construa um programa para calcular as áreas de 10 círculos tendo como dado de entrada o valor de cada raio. Imprimir a maior área calculada.

Escreva uma função para calcular a área de um círculo, dado o seu raio.

Lista 1, exercício 14:

Escreva um programa para imprimir os números primos compreendidos em um intervalo [A .. B], com A < B, que são fornecidos pelo teclado.

Escreva uma função para determinar se um número é primo.

Para o trabalho prático:

Escreva a função `main()` do seu trabalho prático. Ela deve apresentar o menu de opções, ler a opção escolhida e, para as opções 1 a 4, deve chamar o procedimento adequado (um para novo jogo, um para mostrar histórico, um para mostrar recordes e um último para limpar recordes).

Com o que foi aprendido até o momento, já é possível fazer o procedimento de novo jogo (sem armazenar histórico ou recorde). Os demais procedimentos você pode deixar vazios ou incluir um `printf()` dizendo que a funcionalidade ainda não foi implementada.