**UEMG – Universidade do Estado de Minas Gerais – Unidade Frutal**

**Curso – Sistemas de Informação, 1º Período, noturno**

**Disciplina – Introdução à programação**

**Docente – Prof. Sergio Portari**

**Discentes – Estevam Gelain Anselmo**

**Trabalho – Funções**

**Data – 13/08/2021**

**Protótipo de funções**

Nas linguagens C e C++ os protótipos de funções é uma declaração de função que o seu corpo pode ser integre com apenas o seu nome ou seja as funções vão estar na parte de cima do programa e toda vez que você quiser executá-las basta colocar o nome das mesmas, quantas vezes forem necessárias, podemos dizer que com isso o programa pode ficar mais organizado, facilitando até na identificação de erros e em sua manutenção. A seguir uma estrutura básica de declaração de função:

#include <stdio.h>

void texto()

{

printf("\n SUBROTINA.\n");

}

int main()

{

printf("Hello Word");

printf("\n Segunda linha:");

texto();

printf("Programa principal");

return 0;

}

Podemos perceber que o “main” (programa principal) segue lido linearmente sendo interrompido quando a função “texto” foi inserida e executada. A declaração dessa função foi realizada no começo do programa no espaço “void” onde precisamos colocar o nome da sub-rotina na frente e em seguida o que deseja realizar dentro dela.

Em uma sub-rotina podemos realizar tudo que o “main” realiza, podemos declarar novas variáveis, mas elas só poderão ser utilizadas dentro da sub-rotina ela não pode ser utilizada no “main”, podemos também escrever e ler dados e até mesmo chamar outra sub-rotina dentro dela. O exemplo a seguir pode destacar o uso de uma um bloco antes do principal programa e depois sendo chamado na função main. Nele podemos destacar o uso de variáveis, entrada e saída e até mesmo uma operação.

#include <stdio.h>

#include <locale.h>

void fatorial()

{

float x, fat=1;

setlocale(LC\_ALL, "Portuguese");

printf("\n Digite um valor maior que zero: ");

scanf("%d",&x);

while(x > 0){

fat == fat \* x;

x--;

printf(" O numero fatorial: %f", fat);

}

}

int main()

{

printf("\n\n\n Execução da sub rotina");

fatorial();

printf("\n\n\n termino subrotina ");

return 0;

}

**Utilização de uma função como parâmetro de outra função**

Podemos dizer que os parâmetros nada mais é que variáveis declaradas diretamente no início de uma função A finalidade dos parâmetros é fazer a comunicação entre as funções e a função principal. A passagem de valores entre as funções podemos chamar de passagem de parâmetros.

#include <stdio.h>

int divisao(int N1, int N2) //a sub-rotina recebe N1,N2 e retorna um int

{

int resultado;

resultado = N1 / N2;

return(resultado); //retornando o valor do resultado

}

int main(void)

{

int V1, V2, resultado;

printf("Digite o primeiro valor:");

scanf("%d", &V1);

printf("Digite o segundo valor:");

scanf("%d", &V2);

//chama a função e recebe o retorno

resultado = divisao(V1,V2);// aqui está a variável a ser trocada pelas variáveis declarada a cima na sub-rotina

printf("Resultado = %d\n", resultado);

return 0;

}

Ao executar podemos perceber que o “main” vem primeiro e recebe os dados e logo depois que se chama a função e informa o que deseja substituir ela substitui e faz o procedimento que está descrito no void.

**Função recursiva**

Recursividade ou recursão quando uma função chama a si mesma. Em uma função recursiva, a cada chamada é criada na memória uma nova ocorrência da função com comandos e variáveis “isolados” das ocorrências anteriores. A recursão pode ser utilizada sempre que for possível expressar a solução de um problema em função do próprio problema.

Quando uma função é chamada para resolver a sí mesma, podemos dar o exemplo de um numero fatorial.

#include<stdio.h>

int fatorialrec(int num)

{

  if (num == 0) {

    return 1; // caso o num seja igual a zero ele vai retornar e não vai executar mais.

  }

  else {

    return num \* fatorialrec(num-1); // aqui podemos ver que o retorno vai ser o num que vai multiplicar pela função.

  }

}

Int main()

{

Float num;

Printf(“Digite um número para descobrir o seu fatorial: ”);

Scanf(“%f”,&num);

Printf(“\n o numero fatorial de %f e: %f”,num, fatorial(num)); // ele vai substituir o dado do main com a variavel da função e mostrar o resultado.

Return 0;

}