

RELATÓRIO DE TRABALHO DE LABORATÓRIO 2

Disciplina: Linguagem de Programação I

Aluno(s): Ítalo Manzine Amaral Duarte Garofalo (20204027)
Rodrigo Di Bernardi (20206021)

OBJETIVOS DO TRABALHO

- Entender como funciona o mecanismo de funções
- Entender a proposta e as formas de utilização de ponteiros
- Aprender os princípios de alocação dinâmica de memória

1. PARTE TEÓRICA

Responda as questões a seguir:

1. Como funciona o mecanismo de funções em C e em quais situações o seu uso é indicado?

R: Uma função é um módulo (sequência de código) criado para resolver uma tarefa simples e bem definida.

As funções são “chamadas” ou “ativadas” ao longo da execução do programa. No momento da chamada é especificado:

- O nome da função.
- Os argumentos – a informação que é necessária para que a função cumpra seu propósito.

Depois da execução a função retorna o resultado para o ponto em qual foi chamada (normalmente isso ocorre usando comando *return*).

O uso de funções é indicado quando o código tenta resolver diversos problemas e acaba ficando muito grande e complexo. Podemos “quebrar” um problema grande em vários problemas menores, obtendo mais clareza no código, ganhar a capacidade de reutilizar o trecho do código da função apenas chamando-a, e também o trecho da função é relativamente mais independente do código que a chamou. Uma função pode modificar variáveis globais ou ponteiros, mas limitando-se aos dados fornecidos pela chamada de função.

2. O que é uma função recursiva?

R: Uma função recursiva, é aquela que chama a si própria, como por exemplo, uma função que calcula o fatorial de um número inteiro.

3. Como funciona alocação dinâmica de memória e o que é uma variável do tipo ponteiro?

R: A alocação dinâmica funciona com espaços alocados durante a execução do programa, o que permite que o espaço em memória seja alocado apenas quando necessário.

A variável do tipo ponteiro é aquela que contém um endereço de memória, um valor, que é um endereço (como &a) é um valor de ponteiro. Quando um ponteiro contém um determinado endereço, dizemos que ele aponta para o endereço de memória.

2. PARTE PRÁTICA

Problema:

Escreva um programa em linguagem de programação C que realiza as seguintes tarefas:

- a) Criar uma estrutura que representa uma entidade e que possui vários atributos (no mínimo três campos).
- b) Fazer leitura de dados e armazenar a informação sobre no mínimo 3 estruturas.

O armazenamento pode ser feito usando variáveis ou usando um vetor de estruturas.

Tanto as variáveis como vetor podem ser alocados de forma estática ou dinâmica.

- c) Gravar os dados das estruturas num arquivo-texto.

R: Link do código; <https://www.onlinegdb.com/rJHuAHyuO>

2.1. Código desenvolvido com a documentação:

Escrever o código em linguagem c e enviar o código **documentado** em arquivo .c no Moodle ou colocar no relatório o link para código compartilhado pelo <https://www.onlinegdb.com/> ou outro site.

Documentação pode ser feita em inglês ou português e deve seguir o padrão disponível pelo link:

<https://www.bgsu.edu/arts-and-sciences/computer-science/cs-documentation/c-plus-plus-program-documentation-guidelines.html>

2.2. Análise dos resultados e restrições de funcionamento do código:

2.2.1. Hardware

Descrever as características do hardware utilizado:

Exemplo:

O código foi executado num computador com seguintes características:

Processador: i3 7100 dual core 3.9 ghz

Memória operativa RAM: 8gb ddr4 2666mhz

Arquitetura: x64

2.2.2. Análise dos resultados

Preparar dados de entrada e fazer uma análise para avaliar se os dados de saída correspondem a expectativa.

Exemplo:

O código foi executado para seguintes casos:

1.

Variável	Valor
nome	Italo
genero	masculino
idade	29
nota	10
num	1

O resultado da execução do código está de acordo com o esperado.

Tempo de execução de cálculos aproximado: menor que 1s

2.

Variável	Valor
nome	Rodrigo
genero	masculino
idade	18
nota	4

num	0
-----	---

O resultado da execução do código esta de acordo com o esperado.
Tempo de execução de cálculos aproximado: menor que 1s

2.2.3. Restrições

Todos os programas possuem algum tipo de limitação, seja pelo tipo de variáveis usadas ou pela lógica do problema. Descreva essas limitações e restrições.

Exemplo:

Valor máximo para variável do tipo **int**: 2 147 483 647

Valor mínimo para variável do tipo **int**: -2 147 483 648

A variável nota não pode ser menor do que 0 e maior do que 10.

A variável num para a escolha de salvar ou não o arquivo esta limitada em 0 e 1.