Engenharia de Telecomunicações
Programação I

19/05/2023

Avaliação de PRG1 : PARTE 2

Professor: Eraldo Silveira e Silva

- Prova individual e sem consulta. Durante a avaliação o professor poderá monitorar a avaliação.
- usar somente ferramentas locais (não usar codespace);
- Enviar pelo SIGAA somente arquivos fontes. Compactar em arquivi único. Siga instrução do professor.

1 Exercício 1 - Valor 7

Considere um programa que possui uma tabela global de alunos chamada *tabela_alunos[]*, onde cada aluno possui 3 avaliações registradas em um vetor de avaliações, conforme mostrado no esqueleto do programa abaixo. Implemente:

- uma função *retorna_aluno()* que recebe como parâmetros um ponteiro para uma tabela de alunos e o nome de um aluno. A função retorna um ponteiro para a estrutura correspondente ao aluno passado ou NULL, caso o aluno não seja encontrado na tabela;
- Uma função *media_aluno()* que recebe que recebe como parâmetro um ponteiro para uma tabela de alunos e o nome de um aluno na tabela. A função deve retornar a média do aluno. Se o aluno não existir deve retornar -1. Esta função DEVE usar a função *retorna_aluno()* desenvolvida acima;
- Uma função *media_turma()* que recebe como parâmetro um ponteiro para uma tabela de alunos e retorna a média de toda a turma. Esta função DEVE usar a função *media_aluno()* desenvolvida acima;
- Uma função *numero_alunos_acima_media* que recebe como parâmetro um ponteiro para uma tabela de alunos e retorna o número de alunos com nota acima da média. A função deve usar as duas funções desenvolvidas acima: *media_turma()* e função *media_aluno()*;
- Testar as quatro funções no *main()* de forma exaustiva. Usar como tabela global a *tabela_alunos[]*.

Ver o esqueleto e saída abaixo:

```c
#include <stdio.h>
#include <string.h>

#define TAM_ID 10
#define TAM_NOME 30
#define TAM_USERS 5

typedef struct {
                    char nome[30];
                    int aval[3];
               } tipo_aluno;

/* Variável Global - Tabela de Usuários do Sistema */
```

```
tipo_aluno tabela_alunos[TAM_USERS] = {
        {"Silvana e Silva", {10, 7, 9}},
        {"Maria Luisa e Silva", {6, 7, 10}},
        {"Vica e Silva",{10, 10, 9}},
        {"Lara e Silva",{10, 7, 9}},
        {"Eraldo e Silva",{3, 2, 5}},
};

tipo_aluno *retorna_aluno(tipo_aluno *pturma, char *nome}
{
  /* retorna um ponteiro para a estrutura que contem o aluno nome */
}

float media_aluno(tipo_aluno *pturma, char *nome)
{
    /* calcula a media do aluno cujo nome
    foi passado como parâmetro
    usar a função retorna_aluno()
    */
}

float media_turma(tipo_aluno *pturma)
{
    /* retornar a media da turma
       usar a função media_aluno()
    */
}

int numero_alunos_acima_media(tipo_aluno *pturma)
{
    /* retornar o número de alunos acima da media - usar as duas funções acima*/
}

int main()
{
    /*testar as funções separadamente e de forma exaustiva
      aqui.
    */
}
```

2 Exercício 2 - Valor 3

Implemente uma função C chamada *num_vogais()* **usando o conceito de ponteiros**. Ela recebe uma *string* como parâmetro e retorna o número de vogais da *string*. Para testar esta função, entre com *strings* pela linha de comando na chamada do programa e chame devidamente a função *num_vogais()* no *main()* tantas vezes quantos forem as *strings* passadas. Ver esqueleto abaixo.

```
#include <stdio.h>


int num_vogais(char *p)
{
  /* a função deve retornar o número de vogais da string passada */
}

int main(int argc, char *argv[] )
{
```

```
  /* implementar os testes aqui, usando strings passadas em linha de comando
}
```

Exemplo: Suponha que o programa seja chamado no *prompt* da forma:

```
$ conta_vogais abacate amora pera
```

A saída poderia ser:

```
abacate: 4 vogais
amora: 3 vogais
pera: 2 vogais
```

OBS: Note o programa deve funcionar para quaisquer número de *strings* passadas. Para tanto, use o parâmetro *argc* do main().