

TAD Teste

Vamos construir um TAD para nos auxiliar nos testes de funções que desenvolveremos durante toda disciplina. A ideia será similar a função `assert()` da biblioteca `assert.h`, porém com alguns ajustes.

Dados

Quais dados precisam ser representados e manipulados?

- Descrição do teste
- Número total de testes
- Número de acertos
- Número de erros

Operações

Quais funcionalidades serão disponibilizadas?

1. Criar o teste
2. Desalocar o teste
3. Realizar o teste
4. Imprimir um resumo dos

Especificação

```
C v
/*****
*   DADOS
*****/
typedef struct teste Teste;

/*****
*   OPERACOES
*****/
Teste* teste_criar(char* descricao);
void teste_verificar(Teste* t, int condicao, char *mensagem);
void teste_relatorio(Teste* t);
void teste_desalocar(Teste** t);
```

Utilização

```

#include <stdio.h>
#include "tad_teste.h"

int main(){

    Teste* t1 = teste_criar("Testes para verificar operadores relacionais");
    teste_verificar(t1, 1==1, "Teste para expressao 1==1");
    teste_verificar(t1, 20>20 == 0, "20>20 == 0");
    teste_verificar(t1, 5<=5, "5<=5");

    Teste* t2 = teste_criar("Testes para verificar operadores aritmeticos");
    teste_verificar(t2, 2*5 == 10, "2*5 == 10");
    teste_verificar(t2, 2+2 == 4, "2+2 == 4");

    teste_relatorio(t1);
    teste_relatorio(t2);

    teste_desalocar(&t1);
    teste_desalocar(&t2);

    return 0;
}

```

A função teste_verificar deve exibir uma mensagem com o seguinte formato:

Teste com resultado verdadeiro

```
[OK]: Mensagem recebida por parametro
```

Teste com resultado negativo

```
Erro: Mensagem recebida por parametro
```

A função teste_relatorio deve exibir uma mensagem com o seguinte formato:

```
-----  
<NOME DO TESTE>  
-----  
Total    = <QUANTIDADE DE TESTES REALIZADOS>  
Passaram = <QUANTIDADE DE TESTES QUE PASSARAM (TRUE)>  
Falharam = <QUANTIDADE DE TESTES QUE FALHARAM (FALSE)>
```

Veja a saída completa do código acima:

```
./main  
Erro: Teste para expressao 1==1  
[OK]: 20>20 == 0  
[OK]: 5<=5  
[OK]: 2*5 == 10  
[OK]: 2+2 == 4  
  
-----  
Testes para verificar operadores relacionais  
-----  
Total    = 3  
Passaram = 2  
Falharam = 1  
  
-----  
Testes para verificar operadores aritmeticos  
-----  
Total    = 2  
Passaram = 2  
Falharam = 0
```

Implementação

Desenvolver as funções especificadas.

Automatização da compilação

makefile

all:

run:

clean: