



1. Crie uma classe Java, chamada **Data**, que poderá ser usada para a representação de datas. A classe deve conter as seguintes propriedades:

- Atributos **dia**, **mês** e **ano**, do tipo **int**;
- Método **public int get(int e)**;
- Método **public String getString()**.

O método **get(int e)** deve retornar o dia (parâmetro **e==1**), o mês (parâmetro **e==2**) ou o ano (parâmetro **e==3**). Para outros valores de **e** o método deve retornar o valor **-1**;

O método **getString()** deve retornar a data no formato **dd/mm/aaaa** (String);

Use o método **Integer.toString(x)**, para converter um inteiro para String, e o operador **+**, para concatenar duas Strings.

Inclua o método **Data()** na classe que você construir:

```
public Data(int d,int m,int a) {  
    dia=d;  
    mes=m;  
    ano=a;  
}
```

Use a classe abaixo para testar o seu código:

```
public class EX0202 {  
    public static void main(String[] args) {  
        Data d=new Data(11,8,2011);  
        String  
meses[]={ "JAN", "FEV", "MAR", "ABR", "MAI", "JUN", "JUL", "AGO", "SET", "  
OUT", "NOV", "DEZ" };  
  
        System.out.println("Data Editada: "+d.getString());  
        System.out.println(d.get(1)+" de "+meses[d.get(2)-  
1]+" de "+d.get(3));  
    }  
}
```

2. Crie um **TAD Data**, em C, que contenha as mesmas funcionalidades da classe Java acima. Acrescente uma função, chama **dtComp**, que receba duas datas e retorne um inteiro que indique se elas são iguais (0), se a 1ª é posterior à 2ª (valor positivo) ou se a 2ª é posterior à 1ª (valor negativo).

Use os arquivos abaixo para testar o seu TAD.

Arquivo Data.h



```
typedef struct data Data;  
Data *dtCria(int d,int m,int a);  
int dtGet(Data *dt,int e);  
char *dtGetString(Data *dt);  
int dtComp(Data *d1,Data *d2);
```

Arquivo Main.cpp

```
#include "Data.h"  
#include <stdio.h>  
  
int main(void) {  
    Data *d1=dtCria(9,6,1961);  
    Data *d2=dtCria(20,8,2018);  
    int r;  
  
    r=dtComp(d1,d2);  
    if(r > 0)  
        printf("%s é posterior a  
%s\n",dtGetString(d2),dtGetString(d1));  
    else  
        if (r < 0)  
            printf("%s é anterior a  
%s\n",dtGetString(d1),dtGetString(d2));  
        else  
            printf("%s e %s sao  
iguais\n",dtGetString(d1),dtGetString(d2));  
}
```

Observação: use a função **sprintf**, existente na biblioteca **cstdio** de C++ (#include<cstdio>) na criação da função ***dtGetString(Data *dt)**. Ela tem por objetivo copiar dados formatados para uma string.

int sprintf (char * str, const char * format, ...);

Cria uma string com o mesmo texto que seria exibido no monitor se a formatação fosse usada na função **printf**. Em vez da exibição no monitor, a função **sprintf** armazena o texto formatado na string apontado pelo parâmetro **str**.

O tamanho da área de memória apontada por **str** deve ser grande o suficiente para acomodar toda a string resultante.

Um caractere nulo de terminação é automaticamente anexado após o conteúdo.

Em caso de sucesso, o número total de caracteres escritos na área apontada por **str** é retornado. Essa contagem não inclui o caractere nulo adicional, acrescentado automaticamente no final da string. Em caso de falha, um número negativo é retornado.