

Fundamentos de Programação 1

Linguagem C

“Union, Enum e Typedef ”

Slides 12

Prof. SIMÃO

Exemplo de Union

```
#include <stdio.h>
#include <stdlib.h>

union NumCarac
{
    int num;
    char carac;
};

int main()
{
    NumCarac teste1, teste2;

    teste1.num = 2;

    teste1.carac = 'D';

    // printf("O numero é: %i \n", teste1.num);

    printf("O caracter é: %c \n", teste1.carac);

    fflush(stdin);
    system("Pause");
    return 0;
}
```

Exemplo de Enum

```
#include <stdio.h>
#include <stdlib.h>
```

```
enum
{
```

```
    janeiro = 1,
    fevereiro,
    marco,
    abril,
    maio,
    junho,
    julho,
    agosto,
    setembro,
    outubro,
    novembro,
    dezembro
```

```
};
```

```
int main()
```

```
{
```

```
    printf ("Janeiro      é %i o. mês do ano \n", janeiro);
    printf ("Fevereiro   é %i o. mês do ano \n", fevereiro);
    printf ("Março        é %i o. mês do ano \n", marco);
    printf ("Abril         é %i o. mês do ano \n", abril);
    printf ("Maio         é %i o. mês do ano \n", maio);
    printf ("Junho        é %i o. mês do ano \n", junho);
    printf ("Julho        é %i o. mês do ano \n", julho);
    printf ("Agosto       é %i o. mês do ano \n", agosto);
    printf ("Setembro     é %i o. mês do ano \n", setembro);
    printf ("Outubro      é %i o. mês do ano \n", outubro);
    printf ("Novembro    é %i o. mês do ano \n", novembro);
    printf ("Dezembro    é %i o. mês do ano \n", dezembro);
```

```
    system("Pause");
    return 0;
```

```
}
```

```
C:\Simao\Simao2007\Disciplinas Minhas\Computação I\Exemplos C\EnumExemplo.exe
Janeiro      ú 1 o. mês do ano
Fevereiro   ú 2 o. mês do ano
Março       ú 3 o. mês do ano
Abril        ú 4 o. mês do ano
Maio         ú 5 o. mês do ano
Junho        ú 6 o. mês do ano
Julho        ú 7 o. mês do ano
Agosto      ú 8 o. mês do ano
Setembro     ú 9 o. mês do ano
Outubro      ú 10 o. mês do ano
Novembro     ú 11 o. mês do ano
Dezembro     ú 12 o. mês do ano
Press any key to continue . . .
```

Atenção:

Estudar *typedef* .

Tipo Definido

typedef

```

#include <stdio.h>
#include <stdlib.h>

#define QUANTIDADE 100

struct Pessoa
{

    int Idade;
    char Nome [100];

};

int main()
{

    typedef int INTEIRO;

    typedef float REAL;

    typedef struct Pessoa CIDADAO;

    CIDADAO ListaC[QUANTIDADE];

    INTEIRO conta    = 0;

    REAL  media    = 0.0;

    INTEIRO i = 0;

```

```

    printf ( "\n" );

    for ( i = 0; i < QUANTIDADE; i = i + 1 )
    {

        printf ( " Digite o nome do %d o. cidadão: \n ", i+1 );
        scanf ( "%s", ListaC[i].Nome );

        printf ( " Digite a idade do %d o. cidadão: \n ", i+1 );
        scanf ( "%d", &ListaC[i].Idade );

        conta = conta + ListaC[i].Idade;

        printf ( "\n" );
    }

    media = (REAL)(conta / quantidade);

    printf ( "A média de idade é: %f anos. \n", media );
    printf ( "\n" );

    system ( "Pause" );

    return 0;

}

```

Exercício

- Elaborar um programa em linguagem C que permita receber as notas de cada aluno de uma turma de 100 alunos. O programa deve ser capaz de:
 - 1) Receber o nome do aluno e as notas das suas duas parciais sendo cada aluno tratado por uma variável (e.g. posição de um vetor) do tipo *aluno* a ser definido. Na verdade, *aluno* será um tipo renomeado por meio de *typedef* a partir de *discente*, que é uma *struct* a ser definida.
 - 2) Calcular a média (*ma*) de cada aluno e informar:
 - “Reprovado” se média (*ma*) menor que 5.0 Isto é $ma \geq 0$ e $ma < 5.0$
 - “Exame” se média (*ma*) entre 5.0 e 6.9 Isto é $ma \geq 5.0$ e $ma \leq 6.9$
 - “Aprovado” se média (*ma*) maior que 6.9 Isto é $ma > 6.9$ e $ma \leq 10.0$
 - 3) Calcular a média geral da turma (*mt*) e informá-la.
 - 4) Informar a maior média (*maior1*) e a 2ª maior média (*maior2*) obtida na turma.
 - 5) Calcular quantos alunos (*qta*) obtiveram algumas das parciais acima da média da turma, bem como quantos alunos (*qtb*) obtiveram média individual acima da média da turma.
- Estas 5 funcionalidades deverão ser tratadas por funções e **não** se poderá utilizar variáveis, vetores ou matrizes globais neste programa. Ainda, a função que trata da 5ª funcionalidade não poderá ‘imprimir’ mensagens. Os resultados desta função deverão ser informados na função principal (*main*) ou em outra função.
- Finalmente, o usuário poderá escolher quais funcionalidades utilizar, selecionando uma por vez ou mesmos todas juntas, podendo ainda repeti-las.