Notas de Aula
Exemplos em C (2)

Routo Terada

www.ime.usp.br/~rt

Depto. C. da Computação - USP

N=8, X=4.4

55	2.2	3.3	7.7	8.8	3.3	4.4	1.1
0	1	2	3	4	5	6	7

BUSCASEQ.C

```
/*  BUSCASEQ.C--Exemplo de Busca Sequencial
 *  Problema: dada uma sequencia de N numeros reais (do tipo float)
 *  e um numero real X, verificar se X ocorre na sequencia */
#include <stdio.h>
#define Nmax 101          /* Numero maximo de elementos em R[] */
void main(){
  int  N,   /* numero de elementos em R[] */
       i;
  float R[Nmax],  /* vetor de N elementos */
        X;       /* elemento a ser procurado em R[] */
  /*
   *  Leitura dos parametros
   */
```

```
printf("\nDigitar o numero de elementos de R[] -> ");
scanf("%d", &N);
printf("\nN = %d\n", N);
if(N>Nmax-1){
  printf("\nNumero maximo de elementos foi excedido\n");
  exit(0);
  } /*end if */
printf("Digitar os elementos de R[] -> ");
for(i=0; i<N; i=i+1)
  scanf("%f", &R[i]);
printf("\n");
for(i=0; i<N; i=i+1)
  printf("R[%d] = %f", i, R[i]);
printf("\nDigitar o elemento X a ser procurado em R[] -> ");
scanf("%f", &X);
printf("\nX = %f", X);
```

N=8, X=4.4

55	2.2	3.3	7.7	8.8	3.3	4.4	1.1	4.4
0	1	2	3	4	5	6	7	8

BUSCASEQ.C

```
/* Busca Sequencial de X em R[], em tempo proporcional a N */
R[N] = X;   /* valor X como ``sentinela'' apos ultimo
                    elemento valido de R[] */
i=0;
while(R[i] != X)
  i= i+1;
/*
 *  Dar resposta final
 */
if( i != N )
  printf("\n--- X = %f ocorre em R[]\n", X);
else
    printf("\n --- X = %f nao ocorre em R[]\n", X);
} /* end main */
```

N=8

55	8.8	3.3	7.7	2.2	9.9	4.4	1.1
0	1	2	3	4	5	6	7

ORDEDIR.C

```
/*  ORDEDIR.C--Exemplo de Ordenacao por Selecao Direta
 *  Problema: dada uma sequencia de N numeros reais (tipo float)
 *  ordena'-la em ordem crescente. */
#include <stdio.h>
#define Nmax 100          /* Numero maximo de elementos em R[] */
void main(){
  int  N,   /* numero de elementos em R[] */
       i, j,
       IndMin; /* indice do minimo temporario */
  float R[Nmax],  /* vetor de N elementos */
        temp;     /* variavel temporaria */
```

```
/*
 *  Leitura dos parametros
 */
printf("\nDigitar o numero de elementos de R[] -> ");
scanf("%d", &N);
printf("\nN = %d\n", N);
if(N>Nmax){
  printf("\nNumero maximo de elementos foi excedido\n");
  exit(0);
  } /*end if */
printf("Digitar os elementos de R[] a serem ordenados -> ");
for(i=0; i<N; i=i+1)
  scanf("%f", &R[i]);
printf("\n");
for(i=0; i<N; i=i+1){
  printf("R[%d] = %f", i, R[i]);
  } /* end for i */
```

N=8

	55	8.8	3.3	7.7	2.2	9.9	4.4	1.1
i=0 IndMin=0	0	j=1	j=2	j=3	j=4	j=5	j=6	j=7 **IndMin=7**
	1.1	8.8	3.3	7.7	2.2	9.9	4.4	5.5
i=1 IndMin=1	0	1	j=2	j=3	j=4 **IndMin=4**	j=5	j=6	j=7

```
/* Ordenacao dos elementos em R[], em tempo proporcional a N*N */
   for(i=0; i<N; i=i+1){
    IndMin = i;        /* indice do Minimo temporario */
    for(j=i+1; j<N; j=j+1){
      if(R[IndMin] > R[j])
        IndMin = j;
      } /* end for j */
    temp = R[IndMin];
    R[IndMin] = R[i];
    R[i] = temp;
    } /* end for i */
```

ORDEDIR.C

	55	8.8	3.3	7.7	2.2	9.9	4.4	1.1
i=0 IndMin=0	0	j=1 **IndMin=1**	j=2	j=3	j=4	j=5	j=6	j=7 **IndMin=7**
	1.1	8.8	3.3	7.7	2.2	9.9	4.4	5.5
i=1 IndMin=1	0	1	j=2	j=3	j=4 **IndMin=4**	j=5	j=6	j=7
	1.1	2.2	3.3	7.7	8.8	9.9	4.4	5.5
i=2 IndMin=2	0	1	2	j=3	j=4	j=5	j=6	j=7
	1.1	2.2	3.3	7.7	8.8	9.9	4.4	5.5
i=3 IndMin=3	0	1	2	3	j=4	j=5	j=6 **IndMin=6**	j=7
	1.1	2.2	3.3	4.4	8.8	9.9	7.7	5.5
i=4 IndMin=4	0	1	2	3	4	j=5	j=6	j=7 **IndMin=7**

```
   /*
    *  Dar resposta final
    */
   printf("\n Elementos de R[], em ordem crescente:\n");
   for(i=0; i<N; i=i+1){
     printf(" R[%d]=%f ", i, R[i]);
     } /* end for i */
} /* end main */
```

```
/*
 *  BUSCABIN.C--Exemplo de Busca Binaria
 *  Problema: dada uma sequencia ordenada de N numeros reais
 *  (tipo float)
 *  em ordem crescente, e um numero real X, verificar se X ocorre na
 *  sequencia.
 */
#include <stdio.h>
#define Nmax 100          /* Numero maximo de elementos em R[] */
void main(){
  int  N,   /* numero de elementos em R[] */
       i,
       Esq, /* indice do elemento mais aa esquerda
               no intervalo de busca em R[] */
       Dir; /* idem aa direita */
  float R[Nmax],  /* vetor de N elementos ja ordenados */
        X;        /* elemento a ser procurado em R[] */
```

1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8
0	1	2	3	4	5	6	7

```
/*
 *  Leitura dos parametros
 */
printf("\nDigitar o numero de elementos de R[] -> ");
scanf("%d", &N);
printf("\nN = %d\n", N);
if(N>Nmax){
  printf("\nNumero maximo de elementos foi excedido\n");
  exit(0);
  } /*end if */
printf("Digitar os elementos de R[] em ORDEM CRESCENTE -> ");
for(i=0; i<N; i=i+1)
  scanf("%f", &R[i]);
printf("\n");
```

1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8
0	1	2	3	4	5	6	7

X=7.7

```
for(i=0; i<N; i=i+1){
  if(i!=0 && R[i-1] > R[i])
    printf("\nOs elementos de R[] nao estao em ordem crescente\n");
  printf("R[%d] = %f", i, R[i]);
  } /* end for i */
 printf("\nDigitar o elemento X a ser procurado em R[] -> ");
 scanf("%f", &X);
 printf("\nX = %f", X);
```

X=7.7

1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8
Esq=0	1	2	i=3	4	5	6	Dir=7
1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8
				Esq=4	i=5		Dir=7
1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8
						Esq=6=i	Dir=7

BUSCABIN.C

```
/* Busca Binaria de X em R[], em tempo proporcional
 * a log N na base 2 */
Esq= 0; Dir= N-1;
i=(Esq+Dir)/2;    /* indice do elem. do "meio"de R[] */
while(Esq <= Dir && R[i] != X){
  if(R[i]<X) Esq = i+1;
  else Dir = i-1;
  i=(Esq+Dir)/2;   /* novo indice do elem. do "meio"de R[] */
  } /* end while */
```

X=2.2

1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8
Esq=0	1	2	i=3	4	5	6	Dir=7
1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8
Esq=0	i=1	Dir=2					
1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8

BUSCABIN.C

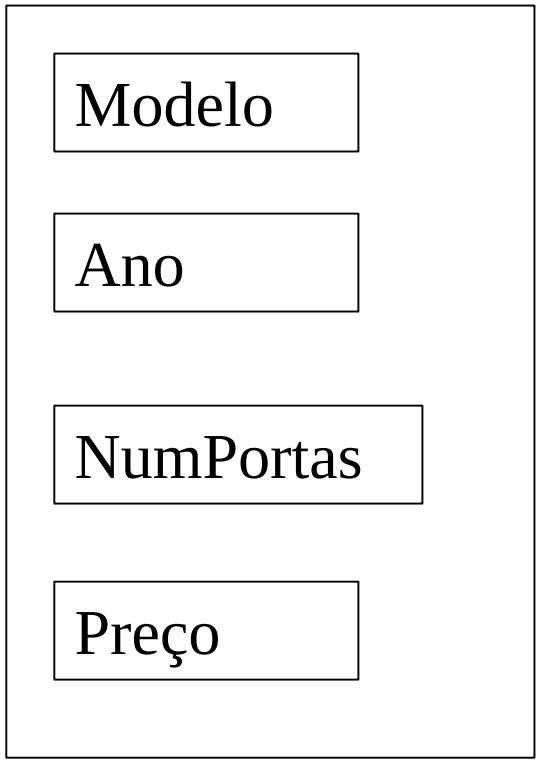
```
/* Busca Binaria de X em R[], em tempo proporcional
 * a log N na base 2 */
Esq= 0; Dir= N-1;
i=(Esq+Dir)/2;    /* indice do elem. do "meio"de R[] */
while(Esq <= Dir && R[i] != X){
  if(R[i]<X) Esq = i+1;
  else Dir = i-1;
  i=(Esq+Dir)/2;   /* novo indice do elem. do "meio"de R[] */
  } /* end while */
```

```
/*
   *  Dar resposta final
   */
  if(R[i] == X)
    printf("\n--- X = %f ocorre em R[]\n", X);
  else
     printf("\n --- X = %f nao ocorre em R[]\n", X);
} /* end main */
```

Grupo de variáveis de tipos distintos

CARRO

struct CARRO



```
// programa de carros em struct
//
#include <stdio.h>
void main(){
 int j;
 struct CARRO {      // CARRO e' o nome do tipo de estrutura
   char *Modelo;
   int Ano;
   int Km;  // quilometragem atual
   char *Fabricante; // nome do fabricante
   char *Cor;
   int NumPortas; // numero de portas
   int GasOuAlc;  // 1==gasolina, 2==alcool
   int Preco;  // preco atual de mercado, em reais
 };   // note o ; aqui
```

CARRO

```
struct CARRO meucarro, carronovo, carrodopai;  // 3 vars do tipo CARRO

meucarro.Modelo= "Astra";
meucarro.Ano= 2000;
meucarro.Km= 31;
meucarro.Fabricante= "GM";
meucarro.Cor= "verde";
meucarro.NumPortas= 2;
meucarro.GasOuAlc= 1;
meucarro.Preco= 32000;
printf("modelo %s\n", meucarro.Modelo);
// sa'ida e' modelo Astra
} // fim main
```

CARRO e Endereço

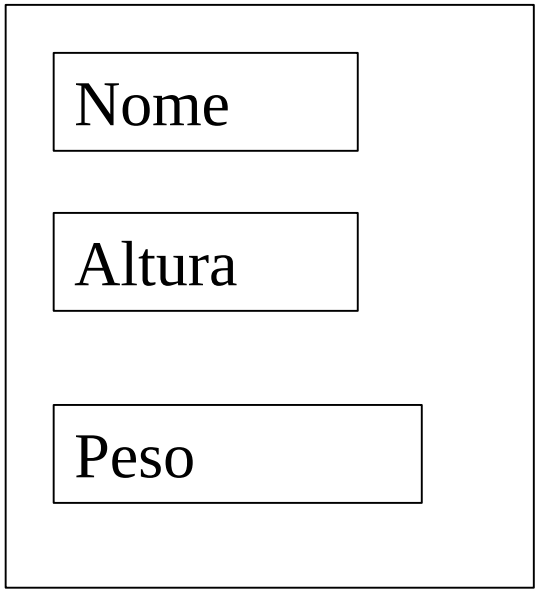
```
// programa de carros em struct
#include <stdio.h>
void main(){
 struct CARRO {      // CARRO e' o nome do tipo de estrutura
   char *NomeDono;
   struct {
     char *RuaeNum; // Nome da rua e numero
     char *Bairro;
     char *CEP;
     char *Cidade;
     char *Telefone;
   } Endereco;     // nome de um componente da estrutura
   char *Modelo;
   int Ano;
   int Km;  // quilometragem atual
   char *Fabricante; // nome do fabricante
   char *Cor;
   int NumPortas; // numero de portas
   int GasOuAlc;  // 1==gasolina, 2==alcool
   int Preco;  // preco atual de mercado, em reais
   char *Chapa;
 };   // note o ; aqui
```

```
struct CARRO meucarro, carronovo, carrodopai;  // 3 vars do tipo CARRO

meucarro.Endereco.RuaeNum= "Rua Padre Anchieta, 1011";
meucarro.Modelo= "Astra";
meucarro.Ano= 2000;
meucarro.Km= 31;
meucarro.Fabricante= "GM";
meucarro.Cor= "verde";
meucarro.NumPortas= 2;
meucarro.GasOuAlc= 1;
meucarro.Preco= 32000;
printf("Nome do meu modelo: %s\n", meucarro.Modelo); // mostra Astra
printf("Nome da minha rua e numero: %s \n", meucarro.Endereco.RuaeNum);

// sa'ida e':
// Nome do meu modelo: Astra
// Nome da minha rua e numero: Rua Padre Anchieta, 1011
}
```

CARRO e Endereço

PESSOA

struct PESSOA




```
// programa de estrutura simples
//
#include <stdio.h>
void main(){

 typedef struct{       // definindo tipo com typedef
   char *Nome;
   int Altura;  // em centimetros
   int Peso;    // em quilos
 } PESSOA;   // nome do tipo aqui
```

```
// note a ausencia da palavra struct nas linhas abaixo
 PESSOA marcio, alcides, maria;  // 3 vars do tipo PESSOA
 PESSOA amigos[20];  // grupo (vetor) de 20 pessoas

 marcio.Altura=170;
 printf("Altura: %d\n", marcio.Altura);

 amigos[2].Peso= 72;
 printf("Peso de um amigo: %d\n", amigos[2].Peso);
 // sa'ida e':
 // Altura: 170
 // Peso de um amigo: 72

}  // fim main
```