

Vetores

Prof. Angelo Loula
UEFS

1

Vetores

- Se o usuário precisa-se digitar uma sequência de números e depois devemos mostrar esta sequência em ordem inversa. Como fazer?

– Para uma sequência de quatro números, quatro variáveis `numero1, numero2, numero3, numero4`

– E para uma sequência de 100 números? Que tal isso?

```
for (X=1; X<=N; X++) {  
    scanf ("%s", &numeroX);  
}
```

– Isto não funciona!!!

2

Vetores

- Como armazenar vários valores em uma única variável?
 - Vetor
 - vários valores em posições diferentes
 - qualquer tipo de elemento, mas todos do mesmo tipo
 - com ordem específica entre si
 - elementos podem ser repetidos ou não

3

Vetores

– Declaração

```
tipo nomevariável[tamanho];
```

– Exemplos

```
int lista[10];
```

```
char respostas[5];
```

```
float numeros[100];
```

4

Vetores

- Utilização
nomevetor[posicao+elemento]
Obs: posição vai de 0 até o tamanho-1
- Exemplos

```
scanf ("%d", &lista[0]);  
printf ("%d", lista[5]);  
printf ("%d", lista[x]);  
lista[5]=lista[x+2];
```

Não é possível fazer: `scanf ("%d", &lista);`

Vetores

```
int lista[10];
```

lista										
posição	0	1	2	3	4	5	6	7	8	9
valor										

```
lista[0]=50;  
lista[1]=100;  
lista[7]=-4;  
lista[2]=0;
```

lista										
posição	0	1	2	3	4	5	6	7	8	9
valor	50	100	0						-4	

Vetores

- Varrendo todos elementos

```
int lista[10];  
for(i=0; i<10; i++){  
    scanf ("%d", &lista[i]);  
}  
for(i=0; i<10; i++){  
    printf ("%d", lista[i]);  
}  
for(i=0; i<10; i++){  
    lista[i]=2*i-1;  
}
```

Vetores

- Vetores
 - Tamanho do vetor é fixo, não é possível alterar.
 - É preciso prever a quantidade máxima de elementos.
 - Podemos manter uma variável numérica para sabermos quantas posições estão sendo usada de fato.
 - Não devemos acessar posições fora dos limites.
 - Programa iria acessar local inválido da memória.
 - Erro fatal (operação ilegal, *segmentation fault*) pode levar o programa a ser 'finalizado' pelo sistema operacional.

```

#define QUANT 10
int main(){
    float nota [QUANT];
    float soma, media;
    int i;

    soma = 0;
    for(i=0; i<QUANT; i++){
        printf("Entre com a nota %d:",i);
        scanf("%f",&nota[i]);
        soma = soma+ nota[i];
    }
    media = soma/QUANT;
    printf("A média das notas é %f",media);
    for(i=0; i<QUANT; i++){
        if (nota[i] > media){
            printf("%f\n",nota[i]);
        }
    }
}

```

9

10

```

#define MAX 100;
int main(){
    int n,i, Soma;
    int v[MAX];
    float M;

    scanf("%d",&n); /*Leitura da quantidade de elementos*/
    while (n > MAX){ /*Quantidade inválida*/
        scanf("%d", &n);
    }
    for(i=0; i<n; i++){
        scanf("%d",&v[i]);
    }
    soma = 0;
    for(i=0; i<n; i++){
        Soma = Soma + v[i];
    }
    M = Soma/n;
    for(i=0; i<n; i++){
        if (v[i] > M) {
            printf("%d",v[i]);
        }
    }
}

```

```

int main() { /* Busca Sequencial */
    var
    int Vmat[10];
    float Vnota[10];
    int i,mat;

    for(i=0; i<10; i++){
        printf("Entre com o matrícula e nota: ");
        scanf("%d %f", &Vmat[i], &Vnota[i]);
    }
    printf("Entre com a matrícula para pesquisar: ");
    scanf("%d",&mat);
    i=0;
    while ((Vmat[i]!=mat) && (i<10)){
        i++;
    }
    if (i<10){
        printf("A nota para %d eh %f",Vmat[i], Vnota[i]);
    }
}

```

11

12

```

#define MAX 100;
int main(){
    int i,n;
    int v[MAX];

    scanf("%d",&n); /* 0 <= n <= TamMax */
    for(i=0; i<n; i++){
        scanf("%d",&v[i]);
    }
    for(i=n-1; i>=0; i--){
        printf("%d",v[i]);
    }
}

```