

**Lista de Exercícios 2 – Algoritmos e Estruturas de Dados I**  
**Prof. Rafael**  
**Fernandes Lopes**  
**Comandos de Atribuição, comando SE e comando**  
**REPITA, ENQUANTO E PARA**

**Adriano dos Santos elizeu**

1 Dado um vetor qualquer com 100 números reais, faça um programa que informa se há ou não números repetidos nesse vetor.

```
#include <stdio.h>
#define LIM 100
int main(){
    float n[LIM];
    float rep[LIM];
    int i,j, cont=0;
    for(i = 0; i < LIM; i++){
        printf("Entre com um numero: ");
        scanf("%f", &n[i]);
    }
    for(i = 0; i < LIM; i++){
        for(j = i + 1; j < LIM; j++){
            if(n[i]==n[j]){
                cont++;
            }
        }
    }

    if (cont > 0){
        printf ("\n=== Há números repetidos ==");
    }
    return 0;
}
```

2 Crie um programa que imprime na tela os elementos de um vetor de 35 números inteiros na ordem inversa.

```
#include <stdio.h>
#define LIM 35
int main (){
    int num[LIM], i;
    for(i = 0; i < LIM ; i++){
        printf("Entre com número: ");
        scanf("%d", &num[i]);
    }
}
```

```

printf("\n::::: ORDEM INVERSA :::::\n");
for(i = 35 ; i >= 0 ; i--){
printf("%d\n", num[i]);

}
return 0;

```

3 Escreva um algoritmo de ordenação para um vetor de N ( $1 \leq N \leq 100$ ) elementos

```

#include <stdio.h>
#define LIM 100
int main (){
int num[LIM], i, j, ordem;
for(i = 0; i < LIM ; i++){
printf("Entre com número: ");
scanf("%d", &num[i]);
}
for (i = 0; i < LIM ; i++){
for(j = i + 1 ; j < LIM; j++ ){
if ( num[i] > num[j]){
ordem = num[i];
num[i] = num[j];
num[j] = ordem;
}
}
}
printf("\n::::: ORDEM:::::\n");
for (i = 0; i < LIM; i++){
printf("%d", num[i]);
printf("\n");
}
return 0;
}

```

4 Crie um programa que informa qual o menor número de um vetor de 40 de números inteiros.

```

#include <stdio.h>
#define LIM 40
int main (){
int num[LIM], i, menor;
for(i = 0; i < LIM ; i++){
printf("Entre com número: ");
scanf("%d", &num[i]);
}

```

```

    menor = num[0];
    for(i = 0; i < LIM ; i++){
        if (menor > num[i]){
            menor = num[i];
        }
    }
    printf("\n:::: Menor é: %d", menor);
    return 0;
}

```

5 Dados dois vetores de 30 números inteiros, informe quantos números repetem-se em ambos os vetores.

```

#include <stdio.h>
#define LIM 30
int main(){
    int n1[LIM], n2[LIM];
    int i,j,x,y, cont=0;
    for(i = 0; i < LIM; i++){
        printf("Vet 01:: Entre com um numero: ");
        scanf("%d", &n1[i]);
    }
    printf("\n-----\n");
    for(x = 0; x < LIM; x++){
        printf("vet 02:: Entre com um numero: ");
        scanf("%d", &n2[x]);
    }
    for(i = 0; i < LIM; i++){
        for(j = i + 1; j < LIM; j++){
            if(n1[i]==n1[j]){
                cont++;
                break;
            }
        }
    }
    for(x = 0; x < LIM; x++){
        for(y = x + 1; y < LIM; y++){
            if(n2[x]==n2[y]){
                cont++;
                break;
            }
        }
    }

    if (cont > 0){
        printf ("\n=== Há %d números repetidos em ambos os vetores ==",
cont);

```

```

    }
    return 0;
}

```

6 Faça um algoritmo que escreva todos os números múltiplos de 7 entre 1 e N, sendo N um valor introduzido pelo usuário do programa. Por exemplos: 7, 14, 21, 28, 35.

```

#include <stdio.h>
int main (){
    int i, n, x;
    printf("Digite um número maior que 7: ");
    scanf("%d", &n);
    for (i = 1; i < n; i++){
        if(i %7==0){
            x=i;
            printf("\n%d",x);
        }
    }
}

```

8 Escreva um programa que retira todos os números repetidos das primeiras N posições de um vetor em ordem crescente, colocando-os em ordem crescente no final do vetor. Exemplo: Para o vetor {1,2,2,3,4}, a solução é {1,2,3,4,2,3}.

```

#include <stdio.h>
#define LIM 7
int main(){
    int n1[LIM],n2[LIM];
    int i,j, n=0, ordem;
    for(i = 0; i <LIM; i++){
        printf("Entre com um numero: ");
        scanf("%d", &n1[i]);
    }

    for(i = 0; i < LIM; i++){
        for(j = 0; j < n; j++){
            if(n1[i]==n2[j])
                break;
        }
        if(j == n){
            n2[n] = n1[i];
            n++;
        }
    }
    for (i = 0 ; i < n; i++){

```

```
        printf("%d\n",n2[i]);  
    }  
  
    return 0;  
}
```