

Exercício 01:

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

int main(){

    int indice, valor;
    int vet[5] = {10,47,53,108,953};

    printf("Informe o Indice para a multiplicação: ");
    scanf("%d", &indice);

    valor = vet[indice-1] * 10;

    printf("O valor é %d\n", valor);
}
```

Exercício 02:

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

int main(){

    int indice, valor;
    int n_linha = 5, n_coluna=4;
    int mat[n_linha][n_coluna];

    for(int i=0;i<n_linha;i++){
        for(int j=0;j<n_coluna;j++){
            printf("Informe o valor na posição %d x %d : ", i,j);
            scanf("%d", &mat[i][j]);
        }
    }

    for(int i=0;i<n_linha;i++){
        printf("\n");
        for(int j=0;j<n_coluna;j++){
            printf("%d", mat[i][j]);
        }
    }
    printf("\n")
}
```

Exercício 03:

Para que o código funcione é necessário antes atribuir o tipo `int` à variável “i” e atribuir tipo e valor à variável “n”. Com relação ao código em si, seu é atribuída a um ponteiro “t1” uma alocação de memória de 400 bytes ($50 * \text{sizeof}(\text{double})$). Em seguida, o ponteiro “t2” recebe a mesma alocação de “t1” e o espaço alocado por “t1” é liberado pela função `free(t1)`. Como cada variável `int` possui 4 bytes de tamanho, é possível alocar aproximadamente 100 valores `int` no vetor de 400 bytes criado como demonstrado no código abaixo:

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

int main(){

    int *t1, *t2;
    t1 = malloc(50 * sizeof(double));
    t2 = t1;
    int teste = 50* sizeof(double);
    for(int i =0; i<103; i++){
        t2[i] = i*i;
    }

    for(int i =0; i<103; i++){
        printf("%d ", t2[i]);
    }
    printf("\n%d\n", t1);
    printf("%d\n", teste);
    printf("%d\n", sizeof(t1));
    printf("\n");
    free(t1);
}
```

Obs: 103 foi o limite de repetições, com o valor de 104 já ocorre um erro.