



```
1 //EXERCICIO 1
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 int main()
7 {
8     int *v, tam_v;
9
10    printf("Digite o tamanho do vetor: ");
11    scanf("%d", &tam_v);
12
13    v = (int *)malloc(tam_v * sizeof(int));
14
15    for (int cont = 0; cont < tam_v; cont++)
16    {
17        printf("Digite o valor da posicao %d: ", cont + 1);
18        scanf("%d", &v[cont]);
19    }
20
21    for (int cont = 0; cont < tam_v; cont++)
22    {
23        printf("\nValor da posicao %d -> %d ", cont + 1, v[cont]);
24    }
25
26    free(v);
27    return 0;
28 }
```



```
1 //EXERCICIO 2
2
3 #include <stdio.h>
4 #include <string.h>
5 #include <stdlib.h>
6
7 int main()
8 {
9     char *s, *vogais = "aeiouAEIOU";
10    ;
11    int tam_v;
12
13    printf("Digite tamanho da string: ");
14    scanf("%d", &tam_v);
15
16    getchar();
17
18    s = (char *)malloc(tam_v * sizeof(char));
19
20    printf("Digite a String: ");
21    gets(s);
22
23    for (int cont = 0; cont < tam_v; ++cont)
24    {
25        if (!(strchr(vogais, s[cont])))
26        {
27            printf("%c", s[cont]);
28        }
29    }
30
31    free(s);
32    return 0;
33 }
```



```
1 //EXERCICIO 3
2
3 #include <stdio.h>
4 #include <time.h>
5 #include <stdlib.h>
6
7 int main()
8 {
9     int *v, tam_v;
10
11     printf("Digite o tamanho do vetor: ");
12     scanf("%d", &tam_v);
13
14     v = (int *)malloc(tam_v * sizeof(int));
15
16     srand(time(NULL));
17     for (int cont = 0; cont < tam_v; cont++)
18     {
19         v[cont] = rand() % tam_v + 1;
20     }
21
22     printf("PAR: ");
23     for (int cont = 0; cont < tam_v; cont++)
24     {
25         if (v[cont] % 2 == 0)
26         {
27             printf("%d, ", v[cont]);
28         }
29     }
30
31     printf("\n");
32
33     printf("IMPAR: ");
34     for (int cont = 0; cont < tam_v; cont++)
35     {
36         if (v[cont] % 2 != 0)
37         {
38             printf("%d, ", v[cont]);
39         }
40     }
41
42     free(v);
43     return 0;
44 }
```



```
1 //EXERCICIO 4
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 int main()
7 {
8     int N, *v, X;
9     int conta = 0;
10    printf("Digite o tamanho do vetor: ");
11    scanf("%d", &N);
12
13    v = (int *)malloc(N * sizeof(int));
14
15    printf("\nDigite UM VALOR X: ");
16    scanf("%d", &X);
17
18    for (int cont = 0; cont < N; cont++)
19    {
20        printf("Digite o %d valor do vetor: ", cont + 1);
21        getchar();
22        scanf("%d", &v[cont]);
23    }
24
25    printf("\nMultiplos de %d: ", X);
26    for (int cont = 0; cont < N; cont++)
27    {
28        if (v[cont] % X == 0)
29        {
30            conta++;
31            printf("%d, ", v[cont]);
32        }
33    }
34    printf("\nTotal de Multiplos: %d", conta);
35
36    free(v);
37    return 0;
38 }
```



```
1 //EXERCICIO 5
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 typedef struct sAlunos
7 {
8     char nome[50];
9     int matricula;
10    float nota;
11    char endereco[50];
12 } Aluno;
13
14 int main()
15 {
16     Aluno *aluno;
17     int tam_v;
18
19     printf("Digite o tamanho do vetor: ");
20     scanf("%d", &tam_v);
21
22     aluno = (int *)malloc(tam_v * sizeof(int));
23
24     for (int cont = 0; cont < tam_v; cont++)
25     {
26         printf("\nDigite o nome do aluno %d: ", cont + 1);
27         getchar();
28         gets(aluno[cont].nome);
29
30         printf("Digite a matricula: ");
31         getchar();
32         scanf("%d", &aluno[cont].matricula);
33
34         printf("Digite a nota: ");
35         scanf("%f", &aluno[cont].nota);
36
37         printf("Digite a endereco: ");
38         getchar();
39         gets(aluno[cont].endereco);
40     }
41
42     free(aluno);
43     return 0;
44 }
```

```

1 //EXERCICIO 6
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 typedef struct sAlunos
7 {
8     char nome[50];
9     int matricula;
10    float nota;
11    char endereco[50];
12 } Aluno;
13
14 void Cadastrar(Aluno *aluno, int tam_v)
15 {
16     for (int cont = 0; cont < tam_v; cont++)
17     {
18         printf("\nDigite o nome do aluno %d: ", cont + 1);
19         getchar();
20         gets(aluno[cont].nome);
21
22         printf("Digite a matricula: ");
23         getchar();
24         scanf("%d", &aluno[cont].matricula);
25
26         printf("Digite a nota: ");
27         scanf("%f", &aluno[cont].nota);
28
29         printf("Digite a endereco: ");
30         getchar();
31         gets(aluno[cont].endereco);
32     }
33 }
34
35 void Imprimir(Aluno *aluno, int tam_v)
36 {
37     for (int cont = 0; cont < tam_v; cont++)
38     {
39         printf("\n\nNome: %s", aluno[cont].nome);
40         printf("\nMatricula: %d ", aluno[cont].matricula);
41         printf("\nNota: %.2f", aluno[cont].nota);
42         printf("\nEndereco: %s", aluno[cont].endereco);
43     }
44 }
45
46 int MaiorNota(Aluno *aluno, int tam_v)
47 {
48     int maior = 0;
49     for (int cont = 0; cont < tam_v; cont++)
50     {
51         if (aluno[cont].nota > maior)
52         {
53             maior = cont + 1;
54         }
55     }
56     return printf("\n\nIndice Maior Nota: %d", maior);
57 }
58
59 int main()
60 {
61     Aluno *aluno;
62     int tam_v;
63
64     printf("Digite o tamanho do vetor: ");
65     scanf("%d", &tam_v);
66
67     aluno = (Aluno *)malloc(tam_v * sizeof(Aluno));
68
69     Cadastrar(aluno, tam_v);
70     Imprimir(aluno, tam_v);
71     MaiorNota(aluno, tam_v);
72
73     free(aluno);
74     return 0;
75 }

```

```
1 //EXERCICIO 7
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 int main()
7 {
8     int **M, l, c;
9
10    printf("Digite o numero de linhas: ");
11    scanf("%d", &l);
12
13    printf("Digite o numero de colunas: ");
14    scanf("%d", &c);
15
16    M = (int **)malloc(sizeof(int *) * l);
17    if (M == NULL)
18    {
19        printf("Memoria insuficiente.\n");
20        exit(1);
21    }
22
23    for (int i = 0; i < l; i++)
24    {
25        M[i] = (int *)malloc(sizeof(int) * c);
26        if (M[i] == NULL)
27        {
28            printf("Memoria insuficiente.\n");
29            exit(1);
30        }
31    }
32
33    for (int i = 0; i < l; i++)
34    {
35        for (int j = 0; j < c; j++)
36        {
37            M[i][j] = i, j;
38        }
39    }
40
41    for (int i = 0; i < l; i++)
42    {
43        for (int j = 0; j < c; j++)
44        {
45            printf("%d ", M[i][j]);
46        }
47        printf("\n");
48    }
49
50    for (int i = 0; i < l; i++)
51    {
52        free(M[i]);
53    }
54    free(M);
55    return 0;
56 }
```