

Exercício 1

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(int argc, char *argv[]) {
6     char expressao[30];
7     char telefone[30];
8     char c;
9     int tam, i;
10    printf("Palavra: ");
11    scanf("%s", expressao);
12    tam = strlen(expressao);
13    for (i = 0; i < tam; i++) {
14        c = expressao[i];
15        if (c == '1' || c == '0' || c == '-') {
16            telefone[i] = expressao[i];
17        } else if (c == 'A' || c == 'B' || c == 'C') {
18            telefone[i] = '2';
19        } else if (c == 'D' || c == 'E' || c == 'F') {
20            telefone[i] = '3';
21        } else if (c == 'G' || c == 'H' || c == 'I') {
22            telefone[i] = '4';
23        } else if (c == 'J' || c == 'K' || c == 'L') {
24            telefone[i] = '5';
25        } else if (c == 'M' || c == 'N' || c == 'O') {
26            telefone[i] = '6';
27        } else if (c == 'P' || c == 'Q' || c == 'R' || c == 'S') {
28            telefone[i] = '7';
29        } else if (c == 'T' || c == 'U' || c == 'V') {
30            telefone[i] = '8';
31        } else if (c == 'W' || c == 'X' || c == 'Y' || c == 'Z') {
32            telefone[i] = '9';
33        }
34    }
35    telefone[tam] = '\0';
36    printf("Telefone correspondente: %s\n", telefone);
37    return 0;
38 }
```

Exercício 2

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(int argc, char *argv[]) {
6     char palindromo[101];
7     int i, j, size, pal = 1;
8     printf("String: ");
9     scanf("%s", palindromo);
10    size = strlen(palindromo);
11    i = 0;
12    j = size - 1;
13    while (i != j && i < j) {
```

```

14     if (palindromo[i] != palindromo[j]) {
15         pal = 0;
16         break;
17     }
18     i++;
19     j--;
20 }
21 if (pal) {
22     printf("A palavra e um palindromo.\n");
23 } else {
24     printf("A palavra nao e um palindromo.\n");
25 }
26 return 0;
27 }

```

Exercício 3

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main(int argc, char *argv[]) {
6      srand(time(NULL));
7      int matriz[5][5];
8      int transposta[5][5];
9      int i, j;
10     printf("Matriz gerada:\n");
11     for (i = 0; i < 5; i++) {
12         for (j = 0; j < 5; j++) {
13             matriz[i][j] = rand() % 10;
14             printf("%d ", matriz[i][j]);
15         }
16         printf("\n");
17     }
18     printf("\n");
19     printf("Matriz transposta:\n");
20     for (i = 0; i < 5; i++) {
21         for (j = 0; j < 5; j++) {
22             transposta[i][j] = matriz[j][i];
23             printf("%d ", transposta[i][j]);
24         }
25         printf("\n");
26     }
27     return 0;
28 }

```

Exercício 4

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     int tam, i, j, linha_maior, maior, menor, coluna_menor;
6     int matriz[100][100];
7     printf("Tamanho da matriz quadrada: ");
8     scanf("%d", &tam);
9     printf("Forneca os elementos da matriz:\n");
10    for (i = 0; i < tam; i++) {
11        for (j = 0; j < tam; j++) {
12            scanf("%d", &matriz[i][j]);
13        }
14    }
15    // Encontra o maior elemento da matriz, assim como a linha
16    // em que se encontra.
17    maior = matriz[0][0];
18    linha_maior = 0;
19    for (i = 0; i < tam; i++) {
20        for (j = 0; j < tam; j++) {
21            if (matriz[i][j] > maior) {
22                maior = matriz[i][j];
23                linha_maior = i;
24            }
25        }
26    }
27    // Encontra o menor elemento na linha com o maior elemento
28    // da matriz, assim como sua coluna.
29    menor = matriz[linha_maior][0];
30    coluna_menor = 0;
31    for (i = 0; i < tam; i++) {
32        if (matriz[linha_maior][i] < menor) {
33            menor = matriz[linha_maior][i];
34            coluna_menor = i;
35        }
36    }
37    printf("O elemento minimax da matriz e %d, e se encontra na posicao [%d
38    ][%d].\n", menor, linha_maior, coluna_menor);
39    return 0;
}
```

Exercício 5

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(int argc, char *argv[]) {
6     srand(time(NULL));
7     int matriz[100][100];
8     int aux[100], aux2[100];
9     int i, j;
10    for (i = 0; i < 100; i++) {
11        for (j = 0; j < 100; j++) {
12            matriz[i][j] = rand()%51;
13        }
14    }
15    // Imprime a matriz original
16    printf("Matriz original:\n");
17    for (i = 0; i < 100; i++) {
18        for (j = 0; j < 100; j++) {
19            printf("%2d ", matriz[i][j]);
20        }
21        printf("\n");
22    }
23    printf("\n");
24    // Armazena a linha 30 em um vetor auxiliar
25    for (i = 0; i < 100; i++) {
26        aux[i] = matriz[30][i];
27    }
28    // Armazena a coluna 25 em outro vetor auxiliar.
29    for (i = 0; i < 100; i++) {
30        aux2[i] = matriz[i][25];
31    }
32    // Passa a coluna 25 para a linha 30
33    for (i = 0; i < 100; i++) {
34        matriz[30][i] = aux2[i];
35    }
36    // Passa a linha 30 para a coluna 25
37    for (i = 0; i < 100; i++) {
38        matriz[i][25] = aux[i];
39    }
40    // Imprime a matriz resultante
41    printf("Matriz alterada:\n");
42    for (i = 0; i < 100; i++) {
43        for (j = 0; j < 100; j++) {
44            printf("%2d ", matriz[i][j]);
45        }
46        printf("\n");
47    }
48    printf("\n");
49    return 0;
50 }
```