

Problema 1.

```
1  #include <stdio.h>
2
3  int main(){
4      int A[5][5];
5      for(int i=0; i<5; i++){
6          for(int j=0; j<5; j++){
7              if(i == j) A[i][j] = 1;
8              else A[i][j] = 0;
9          }
10     }
11     for(int i=0; i<5; i++){
12         for(int j=0; j<5; j++){
13             printf("%d ", A[i][j]);
14         }
15         printf("\n");
16     }
17     return 0;
18 }
```

```
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ gcc p1.c -o p1
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ ./p1
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ |
```

Problema 2.

```
1  #include <stdio.h>
2
3  double soma_diagonal_superior(double m[5][5]){
4      double soma;
5      for(int i=0; i<5; i++){
6          for(int j=0; j<5; j++){
7              if(i>j) soma += m[i][j];
8          }
9      }
10     return soma;
11 }
12
13 int main(){
14     double m[5][5];
15     for(int i=0; i<5; i++){
16         for(int j=0; j<5; j++){
17             m[i][j] = i + j;
18         }
19     }
20     printf("Soma diagonal superior: %lf\n", soma_diagonal_superior(m));
21 }
```

```
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ gcc p2.c -o p2
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ ./p2
Soma diagonal superior: 40.000000
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ |
```

Problema 3.

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

double** multiplicacao_matrizes(double m[5][5], double n[5][5]){
    double prod = 0;
    double **nova;
    nova = (double**)malloc(5*sizeof(double*));
    for(int i=0; i<5; i++){
        nova[i] = (double*)malloc(5*sizeof(double));
    }
    for(int i=0; i<5; i++){
        for(int j=0; j<5; j++){
            for(int k=0; k<5; k++){
                prod += m[i][k]*n[k][j];
            }
            nova[i][j] = prod;
            prod = 0;
        }
    }
    return nova;
}
```

```

int main(){
    double m[5][5];
    for(int i=0; i<5; i++){
        for(int j=0; j<5; j++){
            scanf("%lf", &m[i][j]);
        }
    }
    double n[5][5];
    for(int i=0; i<5; i++){
        for(int j=0; j<5; j++){
            scanf("%lf", &n[i][j]);
        }
    }

    double **nova;
    nova = multiplicacao_matrizes(m, n);
    for(int i=0; i<5; i++){
        for(int j=0; j<5; j++){
            printf("%.1lf ", nova[i][j]);
        }
        printf("\n");
    }
    for(int i=0; i<5; i++){
        free(nova[i]);
    }
    free(nova);
    return 0;
}

```

```

igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ gcc p3.c -o p3
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ ./p3 < entrada.txt
47.0 65.0 91.0 133.0 207.0
112.0 170.0 246.0 358.0 542.0
60.0 86.0 122.0 178.0 274.0
81.0 109.0 151.0 221.0 347.0
94.0 130.0 182.0 266.0 414.0
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$ |

```

Problema 4.

```

#include "relogio.h"

Relogio* cria_relogio(int hora, int min, int seg){
    Relogio *r = (Relogio*)malloc(sizeof(Relogio));
    r->hora = hora;
    r->min = min;
    r->segundo = seg;
    return r;
}

void incrementa_tempo(Relogio* r, int hora, int min, int seg){
    r->segundo += seg;
    r->min += r->segundo/60;
    r->segundo = r->segundo%60;
    r->min += min;
    r->hora += r->min/60;
    r->min = r->min%60;
    r->hora += hora;
    if(r->hora >= 24){
        r->hora = 0;
        r->min = 0;
        r->segundo = 0;
        printf("Erro! Limite de tempo excedido.\n");
    }
}

```

```

void decrementa_tempo(Relogio* r, int hora, int min, int seg){
    int cont = 0;
    r->segundo -= seg;
    while(r->segundo < 0){
        cont++;
        r->segundo += 60;
    }
    r->min -= cont;
    cont = 0;
    r->min -= min;
    while(r->min < 0){
        cont++;
        r->min += 60;
    }
    r->hora -= cont;
    r->hora -= hora;
    if(r->hora < 0){
        r->hora = 0;
        r->min = 0;
        r->segundo = 0;
        printf("Erro! Limite de tempo excedido.\n");
    }
}

void exibe_horario(Relogio* r){
    printf("%d:%d:%d\n", r->hora, r->min, r->segundo);
}

Relogio* destroi_relogio(Relogio* r){
    free(r);
}

```

```

1  #ifndef RELOGIO
2  #define RELOGIO
3
4  #include <stdio.h>
5  #include <stdlib.h>
6
7  typedef struct {
8      int hora;
9      int min;
10     int segundo;
11 } Relogio;
12
13 Relogio* cria_relogio(int hora, int min, int seg);
14
15 void incrementa_tempo(Relogio* r, int hora, int min, int seg);
16
17 void decrementa_tempo(Relogio* r, int hora, int min, int seg);
18
19 void exibe_horario(Relogio* r);
20
21 Relogio* destroi_relogio(Relogio* r);
22 |
23 #endif

```

```

1  #include "relogio.h"
2
3  int main()
4  {
5      Relogio *r = cria_relogio(10, 20, 30);
6      exibe_horario(r);
7      incrementa_tempo(r, 2, 4, 6);
8      exibe_horario(r);
9      decrementa_tempo(r, 5, 30, 40);
10     exibe_horario(r);
11     destroi_relogio(r);
12 }

```

```

igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$
gcc -g -c relógio.c
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$
gcc relógio.o p4.o -o p4
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$
./p4
10:20:30
12:24:36
6:53:56
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$

```

Problema 5.

```

#include "matriz.h"

Matriz* cria_matriz(int m, int n){
    Matriz *mat = (Matriz*)malloc(sizeof(Matriz));
    mat->m = m;
    mat->n = n;
    mat->matriz = (double**)malloc(m*sizeof(double*));
    for(int i=0; i<m; i++){
        mat->matriz[i] = (double*)malloc(n*sizeof(double));
    }
    for(int i=0; i<m; i++){
        for(int j=0; j<n; j++){
            mat->matriz[i][j] = 0;
        }
    }
    return mat;
}

void multiplicacao_escalar(Matriz* M, double escalar){
    for(int i=0; i<M->m; i++){
        for(int j=0; j<M->n; j++){
            M->matriz[i][j] *= escalar;
        }
    }
}

```

```

void multiplicacao(Matriz* M, Matriz* N){
    double prod = 0;
    for(int i=0; i<M->m; i++){
        for(int j=0; j<M->n; j++){
            for(int k=0; k<N->n; k++){
                prod += M->matriz[i][k] * N->matriz[k][j];
            }
            printf("%.1lf ", prod);
            prod = 0;
        }
        printf("\n");
    }
}

void altera_elemento(Matriz* M, int m, int n, double novo_valor){
    M->matriz[m][n] = novo_valor;
}

Matriz* destroi_matriz(Matriz* M){
    for(int i=0; i<M->m; i++){
        free(M->matriz[i]);
    }
    free(M->matriz);
    free(M);
}

```

```

1  #ifndef MATRIZ
2  #define MATRIZ
3
4  #include <stdio.h>
5  #include <stdlib.h>
6
7  typedef struct {
8      double **matriz;
9      int m;
10     int n;
11 } Matriz;
12
13 Matriz* cria_matriz(int m, int n);
14 void multiplicacao_escalar(Matriz* M, double escalar);
15 void multiplicacao(Matriz* M, Matriz* N);
16 void altera_elemento(Matriz* M, int m, int n, double novo_valor);
17 Matriz* destroi_matriz(Matriz* M);
18
19 #endif
20

```

```

1  #include "matriz.h"
2
3  int main(){
4      Matriz *mat1 = cria_matriz(5, 5);
5      for(int i=0; i<5; i++){
6          for(int j=0; j<5; j++){
7              mat1->matriz[i][j] = i + j;
8          }
9      }
10     Matriz *mat2 = cria_matriz(5, 5);
11     for(int i=0; i<5; i++){
12         for(int j=0; j<5; j++){
13             mat2->matriz[i][j] = i*j;
14         }
15     }
16     multiplicacao(mat1, mat2);
17     multiplicacao_escalar(mat1, 3);
18     printf("\n");
19     for(int i=0; i<5; i++){
20         for(int j=0; j<5; j++){
21             printf("%.1lf ", mat1->matriz[i][j]);
22         }
23         printf("\n");
24     }
25     destroi_matriz(mat1);
26     destroi_matriz(mat2);
27 }

```

```

igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$
gcc -g -c matriz.c
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$
gcc matriz.o p5.c -o p5
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$
./p5
0.0 30.0 60.0 90.0 120.0
0.0 40.0 80.0 120.0 160.0
0.0 50.0 100.0 150.0 200.0
0.0 60.0 120.0 180.0 240.0
0.0 70.0 140.0 210.0 280.0

0.0 3.0 6.0 9.0 12.0
3.0 6.0 9.0 12.0 15.0
6.0 9.0 12.0 15.0 18.0
9.0 12.0 15.0 18.0 21.0
12.0 15.0 18.0 21.0 24.0
igor@igor-Aspire-A315-56:~/Área de Trabalho/repositorio_faculdade/faculdade/lab_aeds1/lista_9_aeds$

```