

André Felipe dos Santos  
Yuri Nicolau Freire  
Myllena Oliveira Faria

QUESTÃO 1

MAT0 11711EBI020		MAT1 11711ECP001		MAT2 11711EBI015
KAN00 3		KAN01 3		KAN02 3
KCUR0 2		KCUR1 3		KCUR2 2
KNUM0 3		KNUM1 2		KNUM2 7

QUESTÃO 2

```
#include <stdio.h>
#include <stdlib.h>
#define N 10
```

```
#define KAN00 3
#define KAN01 3
#define KAN02 3
```

```
#define KCUR0 2
#define KCUR1 3
#define KCUR2 2
```

```
#define KNUM0 3
#define KNUM1 2
#define KNUM2 7
```

```
float media_de_aleatorios(int ID)
{
    int * p = (int *) malloc(N*sizeof(int));
    int i;
    float media = 0.0;
    for(i = 0; i < N; i++)
    {
        p[i] = rand()%9 + 1;
        media += p[i];
    }
    media = media/N;
    free(p);
    return media;
}
```

```
int main()
{
```

```

                                P3 de MTP
int ID0 = (KAN00+KAN01+KAN02)%9 + 1, //1
ID1 = (KCUR0+KCUR1+KCUR2)%9 + 1, //8
ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1; //4
srand(ID0*100+ID1*10+ID2);
printf("1o: %f\n", media_de_aleatorios(ID0));
printf("2o: %f\n", media_de_aleatorios(ID1));
printf("3o: %f\n", media_de_aleatorios(ID2));
return EXIT_SUCCESS;
}

```

### QUESTÃO 3A

```

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

#define KAN00 3
#define KAN01 3
#define KAN02 3

#define KCUR0 2
#define KCUR1 3
#define KCUR2 2

#define KNUM0 3
#define KNUM1 2
#define KNUM2 7

double f(double x)
{
    double y = 0.0;
    double PI = 4.0*atan(1.0);
    int ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1;
    switch(ID2)
    {
        case 0: y = x*x-5*x+6; break;
        case 1: y = 2.0*PI*x; break;
        case 2: y = PI*x*x; break;
        case 3: y = 6*x/PI; break;
        case 4: y = x*x*2.0*PI; break;
        case 5: y = -x*x+5*x-6; break;
        case 6: y = 3.5*x-2.0; break;
        case 7: y = PI*x/2.0; break;
        case 8: y = -PI*x+0.1*x; break;
        case 9: y = 2.0*x-3.0*PI; break;
        default: y = 0.0;
    }
    return y;
}

```

```

}

int main()
{
    int ID0 = (KAN00+KAN01+KAN02)%9 + 1, //1
    ID1 = (KCUR0+KCUR1+KCUR2)%9 + 1, //8
    ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1; //4
    srand(ID0*100+ID1*10+ID2);
    double x, y; int i;
    FILE * arq;
    remove("dados.dat");
    arq = fopen("dados.dat","ab");
    for(i = 0; i < 100; i++)
    {
        x = (double) rand()/RAND_MAX;
        y = f(x);
        fwrite(&y,sizeof(double),1,arq);
    }
    printf("Codigo: %d%d%d\n",ID0,ID1,ID2);
    fclose(arq);
    return EXIT_SUCCESS;
}

```

## QUESTÃO 3B

```

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

```

```

#define KAN00 3
#define KAN01 3
#define KAN02 3

```

```

#define KCUR0 2
#define KCUR1 3
#define KCUR2 2

```

```

#define KNUM0 3
#define KNUM1 2
#define KNUM2 7

```

```

double media(double a, double b, double c)
{
    return (a+b+c)/3;
}

```

```

int main()
{
    int ID0 = (KAN00+KAN01+KAN02)%9 + 1, //1

```

```

ID1 = (KCUR0+KCUR1+KCUR2)%9 + 1, //8
ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1; //4
FILE * arq;
int idA, idB, idC;
double nA, nB, nC;
arq = fopen("dados.dat","rb");
if(arq == NULL)
{
    fprintf(stderr,"Arquivo inexistente!\n");
    return EXIT_FAILURE;
}
switch(ID2)
{
    case 1: idA = 13; idB = 14; idC = 64; break;
    case 2: idA = 21; idB = 42; idC = 84; break;
    case 3: idA = 23; idB = 37; idC = 46; break;
    case 4: idA = 16; idB = 55; idC = 82; break;
    case 5: idA = 9; idB = 33; idC = 76; break;
    case 6: idA = 0; idB = 39; idC = 99; break;
    case 7: idA = 10; idB = 86; idC = 92; break;
    case 8: idA = 17; idB = 61; idC = 92; break;
    case 9: idA = 11; idB = 24; idC = 77; break;
    case 10: idA = 5; idB = 53; idC = 65; break;
    default: idA = idB = idC = 0;
}

fseek(arq, 21*sizeof(double), SEEK_SET);
fread(&nA, sizeof(double), 1, arq);
fseek(arq, 42*sizeof(double), SEEK_SET);
fread(&nB, sizeof(double), 1, arq);
fseek(arq, 84*sizeof(double), SEEK_SET);
fread(&nC, sizeof(double), 1, arq);

fclose(arq);
printf("Matricula: %d%d%d\n",ID0,ID1,ID2);
printf("Media [%lf %lf %lf] = %lf\n",nA,nB,nC,media(nA,nB,nC));
return EXIT_SUCCESS;
}

```

#### QUESTÃO 4

```

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

#define KAN00 3
#define KAN01 3

```

```

#define KAN02 3

#define KCUR0 2
#define KCUR1 3
#define KCUR2 2

#define KNUM0 3
#define KNUM1 2
#define KNUM2 7

typedef
struct Aluno
{
    char nome[256];
    int matricula;
    unsigned int idade;
}Aluno;

void mostrar(Aluno aluno)
{
    printf("> %s: MAT %03d\n: %u anos;\n", aluno.nome, aluno.matricula,
aluno.idade);
}

void gravar(Aluno aluno)
{
    FILE * arq;
    arq = fopen("registro.txt","a");//alterado de "w" para "a"
    fwrite(&(aluno.nome),256,1,arq);
    fwrite(&(aluno.matricula),sizeof(int),1,arq);
    fwrite(&(aluno.idade),sizeof(unsigned int),1,arq);
    fclose(arq);
}

int ler(FILE * arq, Aluno * paluno, unsigned int id)
{
    fseek(arq,id*sizeof(Aluno),SEEK_SET);
    int ok = fread(&(paluno->nome),256,1,arq);
    fread(&(paluno->matricula),sizeof(unsigned int),1,arq);//substituidos
matricula por idade e vice-versa
    fread(&(paluno->idade),sizeof(int),1,arq);
    return ok;
}

void inicia()
{
    remove("registro.txt");
    Aluno aluno;

```

```

    strncpy(aluno.nome,"Oswald",256);
    aluno.matricula = rand()%999 + 1;
    aluno.idade = rand()%11 + 17;
    gravar(aluno);
    strncpy(aluno.nome,"Andre",256);
    aluno.matricula = 20;
    aluno.idade = 18;
    gravar(aluno);
    strncpy(aluno.nome,"Yuri", 256);
    aluno.matricula = 1;
    aluno.idade = 20;
    gravar(aluno); //Estava faltando chamar o "gravar"
    strncpy(aluno.nome,"Myllena", 256);
    aluno.matricula = 15;
    aluno.idade = 17;
    gravar(aluno);
    strncpy(aluno.nome,"Silvia", 256);
    aluno.matricula = rand()%999 + 1;
    aluno.idade = rand()%15 + 17;
    gravar(aluno);
    strncpy(aluno.nome,"Mickey", 256);
    aluno.matricula = rand()%999 + 1;
    aluno.idade = rand()%9 + 17;
    gravar(aluno);
}

int main()
{
    int ID0 = (KAN00+KAN01+KAN02)%9 + 1, //1
    ID1 = (KCUR0+KCUR1+KCUR2)%9 + 1, //8
    ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1; //4
    srand(ID0*100+ID1*10+ID2);
    Aluno aluno;
    FILE * arq;
    unsigned int i;
    inicia();
    arq = fopen("registro.txt","r");
    i = 0;
    while(!feof(arq))
    {
        if(ler(arq, &aluno, i))
            mostrar(aluno);
        i++;
    }
    fclose(arq);
    return EXIT_SUCCESS;
}

```