**Métodos e Técnicas de Programação**

**Prova 3 (2017/2)**

**Prof. Igor Peretta**

**11 e 12/dezembro/2017**

**INSTRUÇÕES**

**MARQUE SUA TURMA:**[ **X**] U (terças-feiras) Entrega:**12/12/2017 ;** [ ] V (segundas-feiras) Entrega:**11/12/2017**

**Registro do Grupo:**

**ID** **Nome Matrícula**

**1 Matheus Venturi 11711EBI030**

**2 Igor Henrique 11621EBI003**

**3 Lorraine Prisley 11621EBI026**

Se houver apenas um discente no grupo, utilize as matrículas 10021EEL034 e 10011EBI075 como complemento. Se o grupo for constituído por dois discentes , utilize a matrícula 10011EBI075 como complemento. Importante: É necessária a entrega de somente UMA prova por grupo, não importando o número de alunos que o constitui. Essa prova poderá ser entregue: • Impressa (incluindo os códigos-fonte ajustados) no dia da entrega na sala 3N313 (pode colocar por baixo da porta); • ou disponibilizada em formato PDF em conta/repositório no site Github e os códigos-fonte (\*.c), devendo informar ao professor o usuário Github e respectivo repositório com prazo para a mesma data da entrega física pelo e-mail: [iperetta@ufu.br](mailto:iperetta@ufu.br). Mais importante: Todos os programas possuem requisitos de personalização; se você não personalizar quaisquer deles, suas respectivas questões serão zeradas (veja cada questão para detalhes). Cálculo das constantes necessárias durante a prova . A partir de cada matrícula de discente no grupo (incluindo as que são complementos, quando cabíveis), o grupo deve chegar às seguintes constantes KANO#,KCUR# e KNUM#, conforme questão 1 a seguir. Código para gerar informações pertinentes à prova (com exemplo):

#include <stdio.h>

int main() {

// trocar pelas matrículas devidas (letras maiúsculas)

char mat[][16] = {"11111ECP999","10021EEL034","10011EBI075"};

char curso[4]; curso[3] = ’\0’;

int KANO[] = {0,0,0}, KCUR[] = {0,0,0}, KNUM[] = {0,0,0};

int i, j;

for(i = 0; i < 3; i++) {

for(j = 0; j < 5; j++)

KANO[i] += mat[i][j] - 48;

for(j = 5; j < 8; j++)

curso[j-5] = mat[i][j];

if(strcmp(curso,"EAU") == 0) KCUR[i] = 1;

if(strcmp(curso,"EBI") == 0) KCUR[i] = 2;

if(strcmp(curso,"ECP") == 0) KCUR[i] = 3;

if(strcmp(curso,"EEL") == 0) KCUR[i] = 4;

if(strcmp(curso,"ETE") == 0) KCUR[i] = 5;

if(KCUR[i] == 0) KCUR[i] = 6;

for(j = 8; j < 11; j++) KNUM[i] += mat[i][j] - 48;

KANO[i] = KANO[i]%9 + 1;

KNUM[i] = KNUM[i]%9 + 1;

}

for(i = 0; i < 3; i++)

printf("KANO%d = %d; KCUR%d = %d; KNUM%d = %d\n",

i, KANO[i], i, KCUR[i], i, KNUM[i]);

return 0;

}

Figura 1: Resultado do exemplo

KANO0 = 6; KCUR0 = 3; KNUM0 = 1

KANO1 = 5; KCUR1 = 4; KNUM1 = 8

KANO2 = 4; KCUR2 = 2; KNUM2 = 4

**QUESTÃO 1**

Preencha o quadro a seguir com as informações obtidas a partir de suas matrículas:

**MAT0: 11711EBI030**

**MAT1: 11621EBI003**

**MAT2: 11621EBI026**

**KANO0: 3**

**KANO1: 3**

**KANO2: 3**

**KCUR0: 2**

**KCUR1: 2**

**KCUR2: 2**

**KNUM0: 4**

**KNUM1: 4**

**KNUM2: 9**

**QUESTÃO 2**

Para esta questão, o código a ser utilizado: Código Q2

**#include <stdio.h>**

**#include <stdlib.h>**

**#define N 10**

**#define KANO0 3**

**#define KANO1 3**

**#define KANO2 3**

**#define KCUR0 2**

**#define KCUR1 2**

**#define KCUR2 2**

**#define KNUM0 4**

**#define KNUM1 4**

**#define KNUM2 9**

**float media\_de\_aleatorios(int ID) {**

**int \* p = (int \*) malloc(N\*sizeof(int));**

**int i;**

**float soma = 0.0;**

**float media= 0.0;**

**for(i = 0; i < N; i++) {**

**p[i] = rand()%9 + 1;**

**soma += p[i];**

**}**

**free(p);**

**media = soma/N;**

**return media;**

**}**

**int main() {**

**int ID0 = (KANO0+KANO1+KANO2)%9 + 1,**

**ID1 = (KCUR0+KCUR1+KCUR2)%9 + 1,**

**ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1;**

**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;**

**}**

**// O erro de coerência é que o programa está informando o somatório dos números ao invés da média. O outro é so por free(p)**

Figura 2: Código para Questão 2 (p3\_Q2.c)

Requisitos de personalização •Linhas 4 a 12: trocar os números pelos equivalentes dos valores de KANO0, KANO1, KANO2, KCUR0, KCUR1, KCUR2, KNUM0, KNUM1 e KNUM2. O que é esperado •Corrijir os bugs do código. São dois problemas, um de coerência do código e outro que arrisca vazar memória.

**QUESTÃO 3**

Para esta questão, são dois os códigos a serem utilizados: Código Q3a

Figura 3: Código (a) para Questão 3 ( p3\_Q3a.c)

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define KANO0 3

#define KANO1 3

#define KANO2 3

#define KCUR0 2

#define KCUR1 2

#define KCUR2 2

#define KNUM0 4

#define KNUM1 4

#define KNUM2 9

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 = (KANO0+KANO1+KANO2)%9 + 1,

ID1 = (KCUR0+KCUR1+KCUR2)%9 + 1,

ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1;

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;

}

Requisitos de personalização

•Linhas 4 a 12: trocar os números pelos equivalentes dos valores de KANO0, KANO1, KANO2, KCUR0, KCUR1, KCUR2, KNUM0, KNUM1 e KNUM2. O que é esperado •Compilar e rodar o programa para gerar automaticamente o arquivo dados.dat necessário para a questão Q1b.Código Q3b

#include <stdio.h>

#include <stdlib.h>

#define KANO0 3

#define KANO1 3

#define KANO2 3

#define KCUR0 2

#define KCUR1 2

#define KCUR2 2

#define KNUM0 4

#define KNUM1 4

#define KNUM2 9

double media(double a, double b, double c) {

return (a+b+c)/3;

}

int main() {

int ID0 = (KANO0+KANO1+KANO2)%9 + 1,

ID1 = (KCUR0+KCUR1+KCUR2)%9 + 1,

ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1;

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,0,SEEK\_SET);

fseek(arq,1\*sizeof(double),SEEK\_SET);

fseek(arq,2\*sizeof(double),SEEK\_SET);

fread(&nA,sizeof(double),1,arq);

fread(&nB,sizeof(double),1,arq);

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;

}

Figura 4: Código (b) para Questão 3 (p3\_Q3b.c) Requisitos de personalização

•Linhas 3 a 11: trocar os números pelos equivalentes dos valores de KANO0, KANO1, KANO2, KCUR0, KCUR1, KCUR2, KNUM0, KNUM1 e KNUM2. O que é esperado •Usando fseek e fread, escreva o trecho de código que irá ler os números que estão armazenados no arquivo em suas posições idA,idB e idC para inicializar Na,nB e nC; lembre-se de que a contagem começa no ’0’ e que temos sizeof(double) bytes para cada número. •O trecho de código a ser escrito deve ser posicionado no meio do código já escrito, a partir e no lugar da linha 43.

**QUESTÃO 4**

Para esta questão, segue o código a ser utilizado:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define KANO0 3

#define KANO1 3

#define KANO2 3

#define KCUR0 2

#define KCUR1 2

#define KCUR2 2

#define KNUM0 4

#define KNUM1 4

#define KNUM2 9

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+");

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);

fread(&(paluno->idade),sizeof(int),1,arq);

return ok;

}

void inicia() {

remove("registro.txt");

Aluno aluno;

strncpy(aluno.nome,"matheus", 256);

aluno.matricula = 030;

aluno.idade = 20;

gravar(aluno);

strncpy(aluno.nome,"igor",256);

aluno.matricula = 003;

aluno.idade = 19;

gravar(aluno);

strncpy(aluno.nome,"lorraine", 256);

aluno.matricula = 026;

aluno.idade = 21;

gravar(aluno);

strncpy(aluno.nome,"Juriemo", 256);

aluno.matricula = 75;

aluno.idade = 12;

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 = (KANO0+KANO1+KANO2)%9 + 1,

ID1 = (KCUR0+KCUR1+KCUR2)%9 + 1,

ID2 = (KNUM0+KNUM1+KNUM2)%9 + 1;

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;

}

Figura 5: Código para Questão 4 (p3\_Q4.c) Código Q4 Requisitos de personalização •Linhas 4 a 12: trocar os números pelos equivalentes dos valores de KANO0, KANO1, KANO2, KCUR0, KCUR1, KCUR2, KNUM0, KNUM1 e KNUM2. •Linhas 51 a 60: trocar as informações (nome, 3 últimos dígitos da matrícula, idade) pelas suas respectivas informações. Se não houverem três integrantes no grupo, comece trocando pelo primeiro e deixe o(s) que sobrar(em). O que é esperado •Existem dois erros, um na função gravar()(a partir da linha 21) e o outro na função ler()(a partir da linha 30) que não impede o programa de ser compilado mas impedem o programa de apresentar informações corretamente. Conserte esses erros. •Após compilado, o resultado esperado do executável é o de imprimir na tela as informações corretas dos seis alunos cadastrados (incluindo as suas informações personalizadas).