| L logo 25 anos.png  **Universidade Luterana do Brasil**  **ULBRA – Campus Torres**  **Pró-Reitoria de Graduação** | | Tipo de atividade:  Prova ( ) Trabalho ( ) ..... ( )  Avaliação: G1 ( ) G2 ( )  Substituição de Grau: G1 ( ) G2 ( ) | |
| --- | --- | --- | --- |
| Curso: | Disciplina: | | Data: |
| Turma: | Professor(a): | | Valor da Avaliação:  Nota: |
| Acadêmico(a): n°: | | |

Atividade em Dupla.

Peso 1

1 — Preencha um vetor com 100 números inteiros de forma randômica. Quando o número randômico for 999 o preenchimento deve ser encerrado.

Após preenchido O Programa deve possibilitar 4 opções:  
1- exibir uma lista os números presentes no vetor, não incluído o 999.

2- imprimir o vetor na ordem inversa.

3- Exibir o maior número

4 - Exibir quantos números foram sorteados.

5-Encerrar o programa.

#include <stdio.h>

#include <stdlib.h>

#include<time.h>

int main()

{

int i=0, vetor[100], ordem,cont=0,cont9=0, maior=0;

srand(time(NULL));

for(i=0;i<100;i++){

vetor[i]=rand()%1000;

if(vetor[i]==999){

cont9++;

break;

}

if(vetor[i]>maior){

maior=vetor[i];

}

cont++;

}

do{

printf("Digite a opcao desejada:\n 1- Exibir a lista dos numeros. \n 2- Exiba o vetor na forma inversa. \n 3-Exibir o maior numero. \n 4-Exibir quantos numeros foram sorteado. \n 5-Encerrar o programa. \n");

scanf("%d%\*c", &ordem);

switch(ordem){

case 1:

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

printf("%d\n", vetor[i]);

}

break;

case 2:

for(i=cont;i>=0;i--){

printf("%d\n", vetor[i]);

}

break;

case 3:

printf("O maior numero e %d\n", maior);

break;

case 4:

printf("foram gerados %d numeros\n", cont);

break;

}

}while(ordem!=5);

return 0;

}

2-Faça um algoritmo que Preencha 4 vetores:

* o primeiro com a nota da primeira prova;
* o segundo com a nota da segunda prova;
* o terceiro com as médias ponderada das 2 primeiras notas;
* O Quarto com o nome do Aluno.

O programa deve ao fim de cada leitura perguntar se o usuário deseja adicionar um novo aluno. Defina o vetor com no máximo 10 alunos, Após adicionados:

O Programa deve possibilitar 4 opções:  
1- Listar todos os alunos aprovados

2 -Listar todos os alunos reprovados

3- Listar todos os alunos com nota maior que uma nota informada pelo usuário.

4-Informar uma Posição e o programa exibir os dados e notas do aluno.

5- Encerrar o programa.

#include<stdio.h>

#include<string.h>

int main(){

float notas1[10];

float notas2[10];

float medias[10];

char nomes[10][15];

char aprovados[10][10];

char reprovados[10][10];

int aprov = 0, reprov = 0;

int cont = 0, op;

int funcao, posicao, i=0;

float compare;

for(i=0;i<5;i++){

printf("Digite a primeira nota: \n");

scanf("%f%\*c", &notas1[i]);

printf("Digite a segunda nota: \n");

scanf("%f%\*c", &notas2[i]);

medias[i] = (notas1[i]+notas2[i])/2;

printf("Digite o nome do aluno:\n");

gets(nomes[i]);

if(medias[i]>=6){

strcpy(aprovados[aprov], nomes[i]);

aprov++;

}else{

strcpy(reprovados[reprov], nomes[i]);

reprov++;

}

printf("Digite:\n 1-Para adicionar novos alunos. \n 2-Ir para a proxima parte.\n");

scanf("%d%\*c", &op);

if(op==2){

break;

}

cont++;

}

do{

printf("\n\nDigite:\n 1-Listar todos os alunos aprovados. \n 2-Listar todos os alunos reprovados. \n 3-Listar todos os alunos com a nota maior que a nota informada pelo usuario. \n 4-Informar uma posicao e o programa exibir os dados do aluno. \n 5-Encerrar o programa. \n");

scanf("%d%\*c", &funcao);

switch(funcao){

case 1:

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

printf("Aluno %s aprovado.\n", aprovados[i]);

}

break;

case 2:

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

printf("Aluno %s reprovado.\n", reprovados[i]);

}

break;

case 3:

printf("Digite a nota de comparacao: \n");

scanf("%f%\*c", &compare);

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

if(medias[i]>compare){

printf("O aluno %s tem a nota maior que o comparativo.\n", nomes[i]);

}

}

break;

case 4:

printf("Digite a posicao desejada: \n");

scanf("%d%\*c", &posicao);

if(posicao<=cont){

printf("O aluno %s com a primara nota de %.2f com segunda nota de %.2f com media de %.2f.\n", nomes[posicao], notas1[posicao], notas2[posicao], medias[posicao]);

}else{

printf("Posicao invalida");

}

break;

}

}while(funcao!=5);

}

3 - Fazer um algorítimo que seja lido uma palavra e imprima:

* A quantidade de vogais que compõe a palavra.
* A quantidade de letras dessa palavra.

#include <stdio.h>

#include <math.h>

#include <string.h>

int main()

{

char palavra[20];

int contador, numVogais = 0;

printf("Digite uma palavra: \n");

scanf("%s",&palavra);

for(contador=0; contador < palavra[contador]; contador++){

if (palavra[contador]== 'A' || palavra[contador]== 'E' || palavra[contador]== 'I' || palavra[contador]== 'O' || palavra[contador]== 'U' || palavra[contador]== 'a' || palavra[contador]== 'e' || palavra[contador]== 'i' || palavra[contador]== 'o' || palavra[contador]== 'u'){

numVogais++;

}

}

printf("Numeros de vogais de %s: %d\n",palavra,numVogais);

printf("A palavra %s possui %d letras",palavra, contador);

return 0;

}