Filipe Tizzo Souza
1236337

Sistemas de Informação
Estrutura de Dados

1)

i=0 , temp=0, vetor[0]=9, vetor[9]=0
i=1, temp=1, vetor[1]=8, vetor[8]=1
i=2, temp=2, vetor[2]=7, vetor[7]=2
i=3, temp=3, vetor[3]=6, vetor[6]=3
i=4, temp=4, vetor[4]=5, vetor[5]=4
i=5, temp=4, vetor[5]=4, vetor[4]=5
i=6, temp=3, vetor[6]=6, vetor[3]=3
i=7, temp=2, vetor[7]=7, vetor[2]=2
i=8, temp=1, vetor[8]=8, vetor[1]=1
i=9, temp=0, vetor[9]=9, vetor[0]=0

Filipe Tizzo Souza **Sistemas de Informação**
1236337 **Estrutura de Dados**

2) Feito utilizando **C**

```
#include <stdio.h>

int main()
{
    int valorvet;
    int erro=0;

    printf("digite quantas posições de vetores voce deseja: ");
    scanf("%d",&valorvet);
    printf("\n");
    int vet[valorvet];

    for(int i=0;i<valorvet;i++){

        printf("digite o valor do vetor[%d]: ",i);
        scanf("%d",&vet[i]);
    }
    for(int i=0;i<valorvet;i++){
        if((i!=0)&&(vet[i]<vet[i-1])){
            erro+=1;
            break;
        }
    }
    if(erro>0){
        printf("\n  nao");
    }else{
        printf("\n  sim");
    }

    return 0;
}
```

Filipe Tizzo Souza **Sistemas de Informação**
1236337 **Estrutura de Dados**

3)

```
#include <stdio.h>

char cal_matr(int ln, int cl,char jogador){
    char matrix[3][3];

    if(jogador=='x'){
        for(int i=0;i<3;i++){
                for(int y=0;y<3;y++){
                    if((i==ln)&&(y==cl)){
                        matrix[ln][cl]='X';
                    }
                }
            }
    }
    if(jogador=='o'){
        for(int i=0;i<3;i++){
                for(int y=0;y<3;y++){
                    if((i==ln)&&(y==cl)){
                        matrix[ln][cl]='O';
                    }
                }
            }
    }
    return matrix[ln][cl];
}

int main()
{
    char velha[3][3];
    char jogador;
    int ln,cl; //ln = valor para linha, cl = valor para coluna

    for(int i=0;i<3;i++){
        for(int y=0;y<3;y++){
            velha[i][y]='/';
        }
    }
```

```
   for (int a=0;a<9;a++){

      printf("Informe qual linha o jogador X pretende marcar: ");
      scanf("%d",&ln);
      ln-=1;
      printf("\ninforme qual coluna o jogador X pretende marcar: ");
      scanf("%d",&cl);
      cl-=1;
      jogador='x';

      if((velha[ln][cl]!='O')&&(velha[ln][cl]!='X')){// verificar se a posição esta
vazai

         velha[ln][cl]=cal_matr(ln,cl,jogador);

         for(int i=0;i<3;i++){
            for(int y=0;y<3;y++){
               printf("[%c] ",velha[i][y]);
            }
            printf("\n");
         }

      if((velha[1][0]=='X')&&(velha[1][1]=='X')&&(velha[1][2]=='X')){
                  printf("** Jogador X ganhou **");
                  break;
         }if((velha[0][0]=='X')&&(velha[0][1]=='X')&&(velha[0][2]=='X')){
                  printf("** Jogador X ganhou **");
                  break;
         }if((velha[2][0]=='X')&&(velha[2][1]=='X')&&(velha[2][2]=='X')){
                  printf("** Jogador X ganhou **");
                  break;
         }if((velha[0][0]=='X')&&(velha[2][2]=='X')&&(velha[1][1]=='X')){
                  printf("** Jogador X ganhou **");
                  break;
         }if((velha[2][0]=='X')&&(velha[1][1]=='X')&&(velha[1][2]=='X')){
                  printf("** Jogador X ganhou **");
                  break;
         }if((velha[0][0]=='X')&&(velha[2][0]=='X')&&(velha[1][0]=='X')){
                  printf("** Jogador X ganhou **");
```

```
                printf("** Jogador X ganhou **");
                break;
            }if((velha[0][1]=='X')&&(velha[1][1]=='X')&&(velha[2][1]=='X')){
                printf("** Jogador X ganhou **");
                break;
            }if((velha[2][2]=='X')&&(velha[0][2]=='X')&&(velha[1][2]=='X')){
                printf("** Jogador X ganhou **");
                break;}


        if((velha[1][0]=='O')&&(velha[1][1]=='O')&&(velha[1][2]=='O')){
                printf("** Jogador O ganhou **");
                break;
            }if((velha[0][0]=='O')&&(velha[0][1]=='O')&&(velha[0][2]=='O')){
                printf("** Jogador O ganhou **");
                break;
            }if((velha[2][0]=='O')&&(velha[2][1]=='O')&&(velha[2][2]=='O')){
                printf("** Jogador O ganhou **");
                break;
            }if((velha[0][0]=='O')&&(velha[2][2]=='O')&&(velha[1][1]=='O')){
                printf("** Jogador O ganhou **");
                break;
            }if((velha[2][0]=='O')&&(velha[1][1]=='O')&&(velha[1][2]=='O')){
                printf("** Jogador O ganhou **");
                break;
            }if((velha[0][0]=='O')&&(velha[2][0]=='O')&&(velha[1][0]=='O')){
                printf("** Jogador O ganhou **");
                break;
            }if((velha[0][1]=='O')&&(velha[1][1]=='O')&&(velha[2][1]=='O')){
                printf("** Jogador O ganhou **");
                break;
            }if((velha[2][2]=='O')&&(velha[0][2]=='O')&&(velha[1][2]=='O')){
                printf("** Jogador O ganhou **");
                break;}



        }else{
            printf("\n**Essa posição já foi selecionada...**\n            ** **");
            a--;
        }printf("\n");
```

```
        printf("Informe qual linha o jogador O pretende marcar: ");
           scanf("%d",&ln);
           ln-=1;
           printf("\ninforme qual coluna o jogador O pretende marcar: ");
           scanf("%d",&cl);
           cl-=1;
           jogador='o';

        if((velha[ln][cl]!='O')&&(velha[ln][cl]!='X')){


           velha[ln][cl]=cal_matr(ln,cl,jogador);

           for(int i=0;i<3;i++){
              for(int y=0;y<3;y++){
                 printf("[%c] ",velha[i][y]);
              }
              printf("\n");
           }
// verificar se x gahou
           if((velha[1][0]=='X')&&(velha[1][1]=='X')&&(velha[1][2]=='X')){
                    printf("** Jogador X ganhou **");
                    break;
           }if((velha[0][0]=='X')&&(velha[0][1]=='X')&&(velha[0][2]=='X')){
                    printf("** Jogador X ganhou **");
                    break;
           }if((velha[2][0]=='X')&&(velha[2][1]=='X')&&(velha[2][2]=='X')){
                    printf("** Jogador X ganhou **");
                    break;
           }if((velha[0][0]=='X')&&(velha[2][2]=='X')&&(velha[1][1]=='X')){
                    printf("** Jogador X ganhou **");
                    break;
           }if((velha[2][0]=='X')&&(velha[1][1]=='X')&&(velha[1][2]=='X')){
                    printf("** Jogador X ganhou **");
                    break;
           }if((velha[0][0]=='X')&&(velha[2][0]=='X')&&(velha[1][0]=='X')){
                    printf("** Jogador X ganhou **");
                    break;
           }if((velha[0][1]=='X')&&(velha[1][1]=='X')&&(velha[2][1]=='X')){
```

```
                printf("** Jogador X ganhou **");
                break;
        }if((velha[2][2]=='X')&&(velha[0][2]=='X')&&(velha[1][2]=='X')){
                printf("** Jogador X ganhou **");
                break;}

// verificar se o ganhou
        if((velha[1][0]=='O')&&(velha[1][1]=='O')&&(velha[1][2]=='O')){
                printf("** Jogador O ganhou **");
                break;
        }if((velha[0][0]=='O')&&(velha[0][1]=='O')&&(velha[0][2]=='O')){
                printf("** Jogador O ganhou **");
                break;
        }if((velha[2][0]=='O')&&(velha[2][1]=='O')&&(velha[2][2]=='O')){
                printf("** Jogador O ganhou **");
                break;
        }if((velha[0][0]=='O')&&(velha[2][2]=='O')&&(velha[1][1]=='O')){
                printf("** Jogador O ganhou **");
                break;
        }if((velha[2][0]=='O')&&(velha[1][1]=='O')&&(velha[1][2]=='O')){
                printf("** Jogador O ganhou **");
                break;
        }if((velha[0][0]=='O')&&(velha[2][0]=='O')&&(velha[1][0]=='O')){
                printf("** Jogador O ganhou **");
                break;
        }if((velha[0][1]=='O')&&(velha[1][1]=='O')&&(velha[2][1]=='O')){
                printf("** Jogador O ganhou **");
                break;
        }if((velha[2][2]=='O')&&(velha[0][2]=='O')&&(velha[1][2]=='O')){
                printf("** Jogador O ganhou **");
                break;}
      }else{
         printf("\n**Essa posição já foi selecionada...**\n             ** **");
         a--;
      }printf("\n");
      if(a==8){
         printf("** Velha **");
      }
   }
   return 0;}
```

Filipe Tizzo Souza Sistemas de Informação
1236337 Estrutura de Dados

4)
```
#include <stdio.h>

int main()
{
   int matrix[10][10];
   int maior, x,y;

   for(int i=0;i<10;i++){
      for(int j=0;j<10;j++){
         matrix[i][j]= i*j;
         if(j==7){  //criar diversos números diferentes
            matrix[i][j]*=3;
         }
         printf("%d, ",matrix[i][j]);
         if((i==0)&&(j==0)){
            maior=matrix[i][j];
         }else if(matrix[i][j]>maior){
            maior=matrix[i][j];
            x=i;
            y=j;
         }
      }
      printf("\n");
   }
   printf("\n** O maior número está na posição (%d,%d) = %d
**",x,y,maior);

   return 0;
}
```

5)
```
#include <stdio.h>
int main()
{
   struct bandas{
      int n_integrantes;
      int n_ranking;
```

```
    char nome[30];
    char estilo_m[10];
};

struct bandas banda1;
banda1.n_integrantes = 5;
banda1.n_ranking = 3;
banda1.nome[30]= "ultraje_a_rigor";
banda1.estilo_m[10] = "rock";

printf("numero de integrantes: %d", banda1.n_integrantes);
printf("\nnumero no ranking: %d", banda1.n_ranking);
printf("\nnome: %s", banda1.nome);
printf("\nestilo musical: %s", banda1.estilo_m);

struct bandas banda2;
banda1.n_integrantes = 6;
banda1.n_ranking = 4;
banda1.nome[30]= "mamonas assassinas";
banda1.estilo_m[10] = "rock";

printf("\nnumero de integrantes: %d", banda2.n_integrantes);
printf("\nnumero no ranking: %d", banda2.n_ranking);
printf("\nnome: %s", banda2.nome);
printf("\nestilo musical: %s", banda2.estilo_m);

struct bandas banda3;
banda1.n_integrantes = 6;
banda1.n_ranking = 5;
banda1.nome[30]= "pedra leticia";
banda1.estilo_m[10] = "rock";

printf("\nnumero de integrantes: %d", banda3.n_integrantes);
printf("\nnumero no ranking: %d", banda3.n_ranking);
printf("\nnome: %s", banda3.nome);
printf("\nestilo musical: %s", banda3.estilo_m);

struct bandas banda4;
banda1.n_integrantes = 6;
banda1.n_ranking = 2;
```

```
    banda1.nome[30]= "legiao urbana";
    banda1.estilo_m[10] = "rock";

    printf("\nnumero de integrantes: %d", banda4.n_integrantes);
    printf("\nnumero no ranking: %d", banda4.n_ranking);
    printf("\nnome: %s", banda4.nome);
    printf("\nestilo musical: %s", banda4.estilo_m);

    struct bandas banda5;
    banda1.n_integrantes = 5;
    banda1.n_ranking = 1;
    banda1.nome[30]= "engenheiros do havai";
    banda1.estilo_m[10] = "rock";

    printf("\nnumero de integrantes: %d", banda5.n_integrantes);
    printf("\nnumero no ranking: %d", banda5.n_ranking);
    printf("\nnome: %s", banda5.nome);
    printf("\nestilo musical: %s", banda5.estilo_m);

    return 0;
}
```

6) A estrutura de dados mais adequada seria a Fila, onde o primeiro item a entrar na estrutura é o primeiro a sair (First-In-First-Out). Possibilita a execução de tarefas em ordem sequencial da primeira à última.

7)A estrutura de dados mais adequada seria a Pilha, onde o primeiro item a entrar na estrutura é o último a sair (Last-In-First-Out). Ele saberá qual foi o último evento a ser registrado. Por isso será mais fácil para o sistema desfazer e refazer os últimos passos.

Filipe Tizzo Souza **Sistemas de Informação**
1236337 **Estrutura de Dados**

8)

```
#include <stdio.h>

void mudar(int *a){
   *a=4;
}

int main()
{
   int a=2;
   printf("primeiro valor: %d", a);
   mudar(&a);
   printf("\nsegundo valor: %d", a);
   return 0;
}
```