Trabalho Final CCO-ALG (Caça-palavras).

INTRODUÇÃO.

O projeto tem como objetivo, criar um programa que busque palavras em um caça-palavras em 8 direções diferentes. Tal foi feito usando: Structs(Registros); módulos; array e multidimensional array; ponteiros.

Para alocação de matrizes e vetores foi feito uma alocação responsiva, usando o comando malloc

DESENVOLVIMENTO.

Para receber a grade do caça-palavras, foi utilizado previamente uma função que recebe do usuário o tamanho do caça-palavras e retorna para uma variável. Depois uma função recebe a grade do usuário para criar o caça-palavras.

```
Insira o altura ou comprimento da matriz quadrada:
10

Insira os valores para formar o caça palavras(em vetor):
SAPAACAVAAAAPADAVUUJTAPAAPASONASJAVMVARAJPOATRRASRAOJALORAAAOPAOMOUPSLPAPPMVAUAAOATALSVAPAAOTTRAASOP

 S  A  P  A  A  C  A  V  A  A
 A  A  P  A  D  A  V  U  U  J
 T  A  P  A  A  P  A  S  O  N
 A  S  J  A  V  M  V  A  R  A
 J  P  O  A  T  R  R  A  S  R
 A  O  J  A  L  O  R  A  A  A
 O  P  A  O  M  O  U  P  S  L
 P  A  P  P  M  V  A  U  A  A
 O  A  T  A  L  S  V  A  P  A
 A  O  T  T  R  A  A  S  O  P
```

O usuário, por meio de uma função, define quantas palavras devem ser buscadas pelo programa. Logo inicia uma função para receber essas palavras e alocar em uma matriz em que o tamanho máximo das palavras é o tamanho da matriz.

```
Insira o numero de palavras que precisa encontrar:
7
Escreva a palavra 1 (tamanho maximo=10):SAPATO
Escreva a palavra 2 (tamanho maximo=10):UVA
Escreva a palavra 3 (tamanho maximo=10):LARANJA
Escreva a palavra 4 (tamanho maximo=10):PAO
Escreva a palavra 5 (tamanho maximo=10):AMORA
Escreva a palavra 6 (tamanho maximo=10):SAPO
Escreva a palavra 7 (tamanho maximo=10):SOPA
```

Agora o programa, em posse de todos os parâmetros, inicia uma repetição para percorrer todas as palavras buscando a letra inicial delas para somente assim poder analisar se a palavra realmente está na posição encontrada.

Achando a primeira letra, o programa entra em uma sequência de if(se) que caso todas as posições sejam iguais a 0 ele passa para a próxima verificação de direção. Vale observar que as posições pertencem a um tipo struct nomeado de ROI.

Todos os verificadores de posição são praticamente iguais a menos pela forma como percorrem o caça-palavras, dentro deles tem um for(para) que verifica todas as letras da palavra, a partir do momento que ele finaliza, um if(se) verifica se a variável contadora é igual o tamanho da palavra, se sim então a palavras está naquelas coordenadas e o programa retorna aquela posição, se não o for(para) quebrar antes de encontrar a palavras e então a posição retorna como 0 para todos os sentidos.

Caso a palavra não esteja na posição da primeira inicial encontrada, o programa volta a procurar uma inicial, se encontrar ele repete o processo, se não ele retorna a posição da palavra como 0

Se todos os parâmetros foram colocados de forma correta, o programa retorna para o usuário, a palavra, coluna inicial, linha inicial, coluna final, linha final para todas as palavras(SOPA é um caso onde a palavra não foi encontrada).

```
Insira o numero de palavras que precisa encontrar:
7
Escreva a palavra 1 (tamanho maximo=10):SAPATO
Escreva a palavra 2 (tamanho maximo=10):UVA
Escreva a palavra 3 (tamanho maximo=10):LARANJA
Escreva a palavra 4 (tamanho maximo=10):PAO
Escreva a palavra 5 (tamanho maximo=10):AMORA
Escreva a palavra 6 (tamanho maximo=10):SAPO
Escreva a palavra 7 (tamanho maximo=10):SOPA

 SAPATO:
 [0][0] ate [5][5]
 UVA:
 [7][1] ate [5][1]
 LARANJA:
 [9][6] ate [9][0]
 PAO:
 [1][6] ate [3][6]
 AMORA:
 [3][8] ate [7][4]
 SAPO:
 [8][6] ate [8][9]
 SOPA:
 [0][0] ate [0][0]
```

CONCLUSÃO.

O método utilizado para verificar as palavras no caça-palavras, acaba não sendo o mais rápido e nem o mais compacto, porém provou sua funcionalidade, além de estar modularizado o suficiente para permitir que mudanças possam ser feitas sem dificuldade.

Pelos meus próprios testes, é muito difícil que o usuário quebre o código, a única forma que encontrei de fazer isso é colocando algum caractere na hora em que o código pede ou o tamanho da grade, ou a quantidade de palavras.