Para receber os parâmetros do caça-palavras foram utilizados módulos que recebem valor do usuário. Sendo “recebeTamanhoGrade” que recebe o tamanho da grade e “recebeGrade” que recebe a própria grade. A função “recebeTamanhoGrade” é necessária, pois podemos receber vários tamanhos diferentes de vetor então o tamanho do vetor deve ser responsivo, por esse motivo foi criado a função “alocaMemoriaVetor” que recebe como parâmetro “recebeTamanhoGrade” para criar um vetor do tamanho correto dentro da função “recebeGrade” que essa retorna a grade. Após esse processo de recebimento o vetor é impresso em forma de matriz pela função “printMatriz”.

Para receber as palavras que precisam ser encontradas é necessário 2 funções “recebeQuantidadeDePalavras” que recebe do usuário a quantidade de palavras que serão buscadas e “recebePalavras” para receber cada palavra. A função “recebeQuantidadeDePalavras” tem o mesmo objetivo que “recebeTamanhoGrade”. A função “recebePalavras” recebe como parâmetro “recebeTamanhoGrade” e “recebeQuantidadeDePalavras” e cria uma matriz com as palavras. Esses parâmetros são usados por outra função “alocaMemoriaMatriz” cuja finalidade é criar uma matriz responsiva para as palavras.

Após isso é criado um vetor do tipo “ROI” com tamanho igual a “recebeQuantidadeDePalavras” para guardar a posição de início e fim de cada palavra. O tipo “ROI” foi definido como “typedef struct“.

Um “for” percorre todas as palavras, dentro dele existe a função “localizaPalavra” que recebe como parâmetro uma das palavras, o caça-palavras e o tamanho do caça-palavras. Usa esses parâmetros para descobrir o tamanho da palavra e chama outra função, “verificaPalavra”. A função “verificaPalavra” recebe como parâmetro a palavra, o caça-palavras, o tamanho do caça-palavras e o tamanho da palavra. Quando a primeira letra da palavra for encontrada, essa função chama a função “verificaPalavraPosicao” que vai retornar uma variável tipo “ROI” que essa, caso não seja vazia, será retornada, caso contrário o programa procurará pela próxima letra inicial. Se a função não achar mais nenhuma, ela retorna “ROI” com todos os valores 0.

A função “verificaPalavraPosicao” recebe todos os parâmetros que “verificaPalavra” recebe, e mais a posição no vetor onde se encontrou a primeira letra. Uma sequência de “if” é iniciada. Todos os “if” verificam se as variáveis que guardam a posição inicial e final da palavra estão nulas, se estão, uma função que verifica se a palavra está em determinada direção é chamada se não, a função retorna os valores referente a posição da palavra imediatamente.

As funções que verificam as palavras funcionam em 8 direções sendo alguns exemplos: “verificaHorizontal”, “verificaVertical”, “verificaDiagonalParaBaixo”, etc. A única coisa que diferencia elas é como a posição da próxima letra é tratada. Por padrão todas têm um “for” que contém uma variável contadora, esse “for” pode finalizar em 3 situações, uma das letras não corresponde a palavra, se a variável contadora for maior que o tamanho do vetor ou quando a verificação exceder o limite do vetor. Quando o “for” termina e a variável contadora é igual ao tamanho da palavra, então retorna a posição onde a verificação iniciou e a posição onde a verificação parou.

Assim, logo depois que a função “localizaPalavra” termina, ocorre a impressão da posição.

Por fim temos 2 funções que liberam o espaço na memória criado pelas funções “alocaMemoriaMatriz” e “alocaMemoriaVetor” sendo elas respectivamente “freeMatriz” e “freeVet”.

O método utilizado para verificar as palavras no caça-palavras (as funções que ocorrem dentro de “verificaPalavraPosicao”), talvez não seja o mais eficiente, mas provavelmente é um dos menores possíveis.

Resultados:

Inserindo os valores ela consegue criar uma matriz quadrada

```
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
```

Se os valores estiverem errados, os dois erros mais comuns são:

Caso não haja todos os dígitos para completar a matriz, ela ficará incompleta. Normalmente não funciona pelo fato do C não conseguir comparar valores null em um char.

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

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

 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
```

Caso o tamanho da matriz quadrada for menor que o da matriz. Provavelmente algumas palavras vão se perder, mas irão funcionar.

```
Insira o altura ou comprimento da matriz quadrada:
9

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
```

Inserindo o número de palavras é possível ditar quais palavras serão procuradas.

```
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
```

Se o usuário ultrapassar o tamanho das letras das palavras(definido pelo tamanho da matriz). O programa obviamente não encontrará a palavra.

```
Insira o numero de palavras que precisa encontrar:
2
Escreva a palavra 1 (tamanho maximo=10):SAPAACAVAAA
Escreva a palavra 2 (tamanho maximo=10):SAPAACAVA

 SAPAACAVAAA:
 [0][0] ate [0][0]
 SAPAACAVA:
 [0][0] ate [8][0]
```

Se tudo estiver correto, o programa imprime a posição de cada palavra com início e fim. Se o valor de início e fim for 0, então a palavra não foi encontrada.

```
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]
```