PUZZLE GAME

Function Explanation and Implementations

• main:

➢ **declarations**:

a dynamic array of board (e.g. char** board), a file pointer to the file, number of rows, columns and words, a dynamic array of Word_t type

```
char** board; // a dynamic array of board
FILE* myFile; // file pointer to the file
int rows, cols, nrWords; // number of rows, columns and words
Word_t *words; // a dynamic array of Word_t type

char fileName[STRLEN];
char puzzleInfo[STRLEN];
```

➢ **functionality**

- get the name of the file from the user and open the file

```
printf("Enter name of the text file: ");
scanf("%s", fileName);

myFile = fopen(fileName, "r");
if (myFile == NULL)
{
    perror("Error opening file: ");
    return 0;
}
```

- read number of rows, columns and words

```
fgets(puzzleInfo, STRLEN, myFile);
sscanf(puzzleInfo, "%d %d %d", &rows, &cols, &nrWords);
printf("Game is %d rows x %d cols with %d words\n", rows, cols, nrWords);
```

- dynamically allocate your Word_t type array in the size of words

```
words = (Word_t*)malloc(nrWords * sizeof(Word_t));
```

- call loadTextFile function and assign the returned array to your Word_t type array

```
words = loadTextFile(myFile, nrWords);
```

- call createArray and assign the result to your board

```
board = createArray(rows, cols);
```

- call playGame

```
playGame(board, nrWords, words, rows, cols, 0);
```

- free any dynamically allocated array

```
free(board);
free(words);
```

PUZZLE GAME

• loadTextFile:

➢ Create a temporary Word_t type array and dynamically allocate it using the number of words

```
Word_t *temp = (Word_t *)malloc(nrWords * sizeof(Word_t));
```

➢ For each element of this array, read the direction, row-column starting numbers, word to be solved and its hint from the file.

```
for(int i=0 ; i<nrWords ; i++)
{
    fgets(str, STRLEN, myFile);
    sscanf(str, "%c %d %d %s %[^\n]", &temp[i].direction, &temp[i].x, &temp[i].y,  tempWord, tempClue);

    temp[i].word = (char *)malloc(strlen(tempWord) + 1);
    temp[i].clue = (char *)malloc(strlen(tempClue) + 1);
    strcpy(temp[i].word, tempWord);
    strcpy(temp[i].clue, tempClue);
    temp[i].f = 0;
}
```

• createArray:

➢ Dynamically allocate a temporary board array

```
char** board = (char**)malloc(rows * sizeof(char*));
```

➢ Initialize all elements of the array to ‘#’

```
for(int i = 0; i < rows; i++)
{
    board[i] = (char*)malloc(cols * sizeof(char));
    for(int j=0 ; j<cols ; j++)
        board[i][j] = '#';
}
return board;
```

• findIndex:

➢ Return 1 if the given value is found.

```
for(int i=0 ; i<size ; i++)
{
    if(arr[i] == val)
        return 1;
}
return 0;
```

PUZZLE GAME

• displayBoard:

➢ Traverse the column array and display the column number

```
for(int i=1 ; i<=cols ; i++)
    printf(" %d ",i);
```

➢ Traverse the column array and print dashes as separators

```
for(int i=0 ; i<cols ; i++)
    printf(" --");
```

➢ Print row numbers followed by '|' and print corresponding board element

```
for(int i=0 ; i<cols ; i++)
{
    printf("%d\t|", i+1);
    for(int j=0 ; j<rows ; j++)
        printf(" %c ", myBoard[i][j]);
    printf("\n");
}
```

• isBoardFilled:

➢ If there is any blank in the table (e.g. '_'), return 0, else return 1

```
for(int i = 0 ; i < rows ; i++)
{
    for(int j = 0 ; j < cols ; j++)
    {
        if(myBoard[i][j] == '_') return 0;
    }
}
return 1;
```

PUZZLE GAME

• updateBoard:

➢ This function inserts the element with the index:solve into the board.

➢ If the word is solved, insert the word. Else, put underscores (‘_’) in the length of this word.

```
if(words[solve].f == 1)
{   if(words[solve].direction == 'H')
    {
        for(int i=0 ; i<strlen(words[solve].word) ; i++)
            myBoard[words[solve].x-1][i + words[solve].y-1] = words[solve].word[temp++];
    }
    else if(words[solve].direction == 'V')
    {
        for(int i=0 ; i < strlen(words[solve].word) ; i++)
            myBoard[i+words[solve].x-1][words[solve].y-1] = words[solve].word[temp++];
    }
}
```

```
else
{   if(words[solve].direction=='H')
    {
        for(int i=0 ; i<strlen(words[solve].word) ; i++)
            myBoard[words[solve].x-1][i + words[solve].y-1] = '_';
    }
    else if(words[solve].direction=='V')
    {
        for(int i=0 ; i<strlen(words[solve].word) ; i++)
            myBoard[i+words[solve].x-1][words[solve].y-1]='_';
    }
}
```

• playGame:

➢ Initially, update and display the board.

```
while(count < nrWords)
    updateBoard(myBoard, words, count++);

printf("Current puzzle:\n");
displayBoard(x, y, myBoard);
```

➢ Now, create a loop in which the game is played. Game finishes whenever gameOver reaches to the number of total words or whenever there is no more blanks in the board.

```
while(!isBoardFilled(x, y, myBoard) && gameOver < nrWords)
```

PUZZLE GAME

▪ For each word, display the associated direction, row, column info if it is not solved before

```
while(count < nrWords)
{
    if(!words[count++].f)
    {
        if(words[count-1].direction == 'H')
            printf("%d: Horizontal\t%d\t%d\n", count, words[count-1].x, words[count-1].y);
        else if(words[count-1].direction == 'V')
            printf("%d: Vertical\t%d\t%d\n", count, words[count-1].x, words[count-1].y);
    }
}
```

▪ Ask the user to enter a word number

```
char input[STRLEN];
int num;
do
{
    printf("\nEnter -1 to exit\nWhich word to solve next?: ");
    scanf("%s", input);
    num = checkInputValidity(input, words, nrWords);
}while(num == 0);
```

▪ Check the validity of the input (solved before, or valid in range, etc.)

```
int checkInputValidity(char* input, Word_t* words, int nrWords)
{
    int num = atoi(input);

    if(num == -1) exit(0);

    if (num == 0 && input[0] != '0')
    {
        printf("Input is not an integer");
        return 0;
    }
    if(num < 1 || num > nrWords)
    {
        printf("Input is not in range");
        return 0;
    }
    if(words[num-1].f == 1)
    {
        printf("That word has been solved before");
        return 0;
    }
    return num;
}
```

PUZZLE GAME

▪ Show hint

```
printf("Current hint: %s\n", words[num-1].clue);
```

▪ Get the answer

```
char solution[STRLEN];
scanf(" %s", solution);
```

• If correct, increment gameOver counter, update the board and display the board.

```
if(!strcasecmp(words[num-1].word, solution))
{
    printf("\nCorrect!\n\n");
    gameOver++;
    words[num-1].f = 1;
    updateBoard(myBoard, words, num-1);
}
```

```
printf("Current puzzle:\n");
displayBoard(x, y, myBoard);
```

▪ Check whether the board is filled or not.

```
if(isBoardFilled(x, y, myBoard))
{
    printf("Congratulations! You beat the puzzle!\n\n");
    return;
}
```

PUZZLE GAME

Sample Outputs

puzzle2010.txt

```
PUZZLE GAME

Enter name of the text file: puzzle2010.txt
Game is 5 rows x 5 cols with 7 words
Words and clues are read!
Current puzzle:

          1  2  3  4  5
          -- -- -- -- --
1        | _  _  _  _  #
2        | _  _  #  #  _
3        | _  #  _  _  _
4        | _  #  _  #  _
5        | #  #  _  _  _
         ***************

Ask for hint:
#  Direction   row     col
---------------------------
1: Horizontal   1       1
2: Horizontal   2       1
3: Horizontal   3       3
4: Horizontal   5       3
5: Vertical     1       1
6: Vertical     3       3
7: Vertical     2       5

Enter -1 to exit
Which word to solve next?: 1
Current hint: White liquid produced by the mammals
```

Wrong Answer

```
Enter your solution: water
WRONG ANSWER
Current puzzle:

          1  2  3  4  5
          -- -- -- -- --
1        | _  _  _  _  #
2        | _  _  #  #  _
3        | _  #  _  _  _
4        | _  #  _  #  _
5        | #  #  _  _  _
         ***************
```

PUZZLE GAME

Correct Answer

```
Ask for hint:
#  Direction   row     col
--------------------------
1: Horizontal   1       1
2: Horizontal   2       1
3: Horizontal   3       3
4: Horizontal   5       3
5: Vertical     1       1
6: Vertical     3       3
7: Vertical     2       5

Enter -1 to exit
Which word to solve next?: 1
Current hint: White liquid produced by the mammals
Enter your solution: milk

Correct!

Current puzzle:

          1  2  3  4  5
          -- -- -- -- --
1       | M  I  L  K  #
2       | _  _  #  #  _
3       | _  #  _  _  _
4       | _  #  _  #  _
5       | #  #  _  _  _
        **************
```

PUZZLE GAME

Invalid Inputs: Solved word, input value greater than number of words, input value less than number of words, non-integer input(char, string etc.)

```
Ask for hint:
#  Direction   row     col
--------------------------
2: Horizontal   2       1
3: Horizontal   3       3
4: Horizontal   5       3
5: Vertical     1       1
6: Vertical     3       3
7: Vertical     2       5

Enter -1 to exit
Which word to solve next?: 1
That word has been solved before
Enter -1 to exit
Which word to solve next?: 545
Input is not in range
Enter -1 to exit
Which word to solve next?: -12
Input is not in range
Enter -1 to exit
Which word to solve next?: 4
Current hint: The name of a famous author whose surname is Brown
Enter your solution: Dan

Correct!
```

```
Current puzzle:

          1  2  3  4  5
          -- -- -- -- --
1        | M  I  L  K  #
2        | _  _  #  #  _
3        | _  #  _  _  _
4        | _  #  _  #  _
5        | #  #  D  A  N
         ***************

Ask for hint:
#  Direction   row     col
--------------------------
2: Horizontal   2       1
3: Horizontal   3       3
5: Vertical     1       1
6: Vertical     3       3
7: Vertical     2       5

Enter -1 to exit
Which word to solve next?: A
Input is not an integer
Enter -1 to exit
Which word to solve next?: random
Input is not an integer
Enter -1 to exit
Which word to solve next?: 5
Current hint: A set of cognitive faculties, e.g. consciousness, perception, etc.
Enter your solution:
```

PUZZLE GAME

End of the Game

```
Current puzzle:

          1  2  3  4  5
          -- -- -- -- --
1        | M  I  L  K  #
2        | I  N  #  #  I
3        | N  #  B  U  S
4        | D  #  A  #  B
5        | #  #  D  A  N
         ***************

Congratulations! You beat the puzzle!
```

puzzle2010Long.txt

```
PUZZLE GAME

Enter name of the text file: puzzle2010Long.txt
Game is 6 rows x 6 cols with 14 words
Words and clues are read!
Current puzzle:

          1  2  3  4  5  6
          -- -- -- -- -- --
1        | _  _  #  _  _  _
2        | _  #  _  _  _  #
3        | _  #  _  _  #  #
4        | _  _  _  _  #  #
5        | _  _  #  #  _  _
6        | #  #  #  #  _  _
         ******************

Ask for hint:
#  Direction   row     col
---------------------------
1: Horizontal   1       1
2: Horizontal   1       4
3: Horizontal   2       3
4: Horizontal   3       3
5: Horizontal   4       1
6: Horizontal   5       1
7: Horizontal   5       5
8: Horizontal   6       5
9: Vertical     1       1
10: Vertical    4       2
11: Vertical    2       3
```

PUZZLE GAME

Wrong Answer

```
12: Vertical    1       4
13: Vertical    1       5
14: Vertical    5       5

Enter -1 to exit
Which word to solve next?: 4
Current hint: Abbrev. Nuclear Sclerosis
Enter your solution: NC
WRONG ANSWER
Current puzzle:

          1  2  3  4  5  6
          -- -- -- -- -- --
1       | _  _  #  _  _  _
2       | _  #  _  _  _  #
3       | _  #  _  _  #  #
4       | _  _  _  _  #  #
5       | _  _  #  #  _  _
6       | #  #  #  #  _  _
        ******************
```