ASSIGNMENT-8 (PART1)

1. void bosluklariKaldir(char *satir)
- This function removes spaces, tabs, newlines, and carriage returns from a given string.
Parameters:
    satir: Pointer to the string from which spaces will be removed.

2. void data()
- This function reads data from the file named "input1.txt", removes spaces using the bosluklariKaldir() function, and writes the modified data to the file named "output.txt".
- It then reads data from "output.txt", removes duplicate commas, and writes the corrected data to "lastOutput.txt". 3.

void create_objects()
- This function reads data from the file "lastOutput.txt" and creates objects of type struct scientist.
- It reads the data line by line, parses each line to extract name, surname, age, and field of study, and stores them in the respective fields of the struct scientist objects.

4. void sort_by_age(struct scientist bilim_adamlari[], int kayit_sayisi)
- This function sorts the array of struct scientist objects by age in ascending order.
- It implements the Bubble Sort algorithm to achieve sorting.

5. void sort_by_branch()
- This function sorts the array of struct scientist objects primarily by the field of study (branch) and then by age if branches are the same.
- It implements a modified Bubble Sort algorithm to achieve sorting based on two criteria.

6. void filter_people_by_branch()
- This function filters and prints the information of scientists whose field of study contains the string "SCIENCE".

7. void filter_people_by_profession()
- This function filters and prints the information of scientists whose field of study contains the string "COMPUTERSCIENCE" but does not contain the string "MATHEMATICS".

8. int main()
- This function serves as the entry point of the program.
- It calls the data() function to process input data, create_objects() to create objects from processed data, and filter_people_by_profession() to filter and print information about scientists based on their field of study.
- It returns 0 upon successful execution.

```
****** MENU ******
1. Sort and display all individuals by age
2. Sort and display individuals in the branch by age
3. Show individuals with the branch 'SCIENCE'
4. Show computer scientist who are not mathematicians
5. Exit
Choice :1
Hypatia          nan     35      MATHEMATICS,PHILOSOPHY
Ada      Lovelace        37      COMPUTERSCIENCE
Canan    Dagdeviren      38      PHYSICS,MATERIALSSCIENCE
Grace    Hopper          41      COMPUTERSCIENCE,MATHEMATICS
Alan     Turing          42      COMPUTERSCIENCE,MATHEMATICS
Johnvon          Neumann         54      COMPUTERSCIENCE
Özlem    Tureci          57      MEDICINE,IMMUNOLOGY
Aristo   nan     62      PHILOSOPHY
Gregor   Mendel          62      GENETICS,MATHEMATICS
Mehmet   Oz      63      MEDICINE
Dmitri   Mendeleev       63      CHEMISTRY
Marie    Curie   67      CHEMISTRY,PHYSICS
Janaki   Ammal   67      BOTANY,CYTOGENETICS
Sokrates         nan     70      PHILOSOPHY
Feza     Gursey          71      PHYSICS,MATHEMATICS
Charles          Darwin          73      GENETICS,GEOLOGY
Albert   Einstein        76      PHYSICS
Aziz     Sancar          77      CHEMISTRY,MEDICINE
John     Dalton          78      CHEMISTRY,PHYSICS
Behram   Kursunoglu      81      PHYSICS
Isaac    Newton          84      PHYSICS
Cahit    Arf     87      MATHEMATICS
Asuman   Baytop          95      BOTANY,PHARMACY
```

```
****** MENU ******
1. Sort and display all individuals by age
2. Sort and display individuals in the branch by age
3. Show individuals with the branch 'SCIENCE'
4. Show computer scientist who are not mathematicians
5. Exit
Choice :2
Janaki   Ammal   67      BOTANY,CYTOGENETICS
Asuman   Baytop          95      BOTANY,PHARMACY
Dmitri   Mendeleev       63      CHEMISTRY
Aziz     Sancar          77      CHEMISTRY,MEDICINE
Marie    Curie   67      CHEMISTRY,PHYSICS
John     Dalton          78      CHEMISTRY,PHYSICS
Ada      Lovelace        37      COMPUTERSCIENCE
Johnvon          Neumann         54      COMPUTERSCIENCE
Grace    Hopper          41      COMPUTERSCIENCE,MATHEMATICS
Alan     Turing          42      COMPUTERSCIENCE,MATHEMATICS
Charles          Darwin          73      GENETICS,GEOLOGY
Gregor   Mendel          62      GENETICS,MATHEMATICS
Cahit    Arf     87      MATHEMATICS
Hypatia          nan     35      MATHEMATICS,PHILOSOPHY
Mehmet   Oz      63      MEDICINE
Özlem    Tureci          57      MEDICINE,IMMUNOLOGY
Aristo   nan     62      PHILOSOPHY
Sokrates         nan     70      PHILOSOPHY
Albert   Einstein        76      PHYSICS
Behram   Kursunoglu      81      PHYSICS
Isaac    Newton          84      PHYSICS
Canan    Dagdeviren      38      PHYSICS,MATERIALSSCIENCE
Feza     Gursey          71      PHYSICS,MATHEMATICS
```

```
****** MENU ******
1. Sort and display all individuals by age
2. Sort and display individuals in the branch by age
3. Show individuals with the branch 'SCIENCE'
4. Show computer scientist who are not mathematicians
5. Exit
Choice :3
Ada      Lovelace        37      COMPUTERSCIENCE
Johnvon          Neumann         54      COMPUTERSCIENCE
Grace    Hopper          41      COMPUTERSCIENCE,MATHEMATICS
Alan     Turing          42      COMPUTERSCIENCE,MATHEMATICS
Canan    Dagdeviren      38      PHYSICS,MATERIALSSCIENCE
```

```
****** MENU ******
1. Sort and display all individuals by age
2. Sort and display individuals in the branch by age
3. Show individuals with the branch 'SCIENCE'
4. Show computer scientist who are not mathematicians
5. Exit
Choice :4
Ada      Lovelace        37      COMPUTERSCIENCE
Johnvon          Neumann         54      COMPUTERSCIENCE
```

```
****** MENU ******
1. Sort and display all individuals by age
2. Sort and display individuals in the branch by age
3. Show individuals with the branch 'SCIENCE'
4. Show computer scientist who are not mathematicians
5. Exit
Choice :5
ynalbant@DESKTOP-UV42VUV:~/assignment8$
```

ASSIGNMENT-8 (PART2)

1. ft_strstr()
- This function implements a simplified version of the strstr() function in the C standard library.
- It searches for the first occurrence of a substring within a string.

Parameters:
- str: Pointer to the string to be searched.
- to_find: Pointer to the substring to search for.

Returns:
- If the substring is found, it returns a pointer to the beginning of the first occurrence of the substring in the string. If not found, it returns NULL.

2. search_p1()
- This function searches for a specific pattern (***++++***++++***) horizontally in a 2D array of characters (strs).
- It iterates through each row of the array and checks for the pattern's occurrence.

Parameters:
- strs: 2D array of characters representing the data.
- num_rows: Number of rows in the data array.
- matches: 2D array to store the indices of pattern occurrences.
- num_matches: Pointer to an integer to store the number of pattern occurrences found.

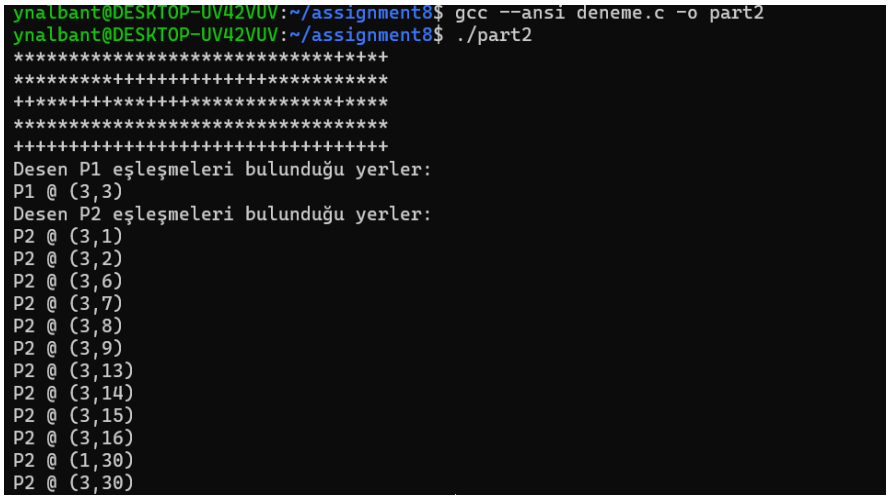
- The function updates the matches array with the row and column indices of each occurrence of the pattern.

3. search_p2()
- This function searches for another specific pattern (+*+*+) vertically in the same 2D array of characters (strs).
- It iterates through each column of the array and checks for the pattern's occurrence vertically.

Parameters and return value are similar to search_p1().

4. get_data()
- This function reads data from a file named "input2.txt" and stores it in the 2D character array data. - It reads each line of the file and removes the newline character before copying it into the array.
- The function prints the data read from the file for verification.

5. main()
- The main() function serves as the entry point of the program.
- It calls get_data() to read data from the file and store it in the data array.
- Then, it calls search_p1() and search_p2() to find occurrences of specific patterns in the data.
- Finally, it prints the locations of pattern occurrences found by both functions.



Youtube Link :
https://youtu.be/4tHN0591uz8