VISHAL G Advanced C Module 1 Assignment

1. C program that represent a calendar for a week. Each day has dayName and tasks

Code:

#include <stdio.h>

#include <string.h>

#include <stdbool.h>

struct tasks

{

    char day[30];

    char tasks[3][40];

};

void print\_list(struct tasks newtask[7]){

    for(int i = 0; i < 7; i++){

        printf("%s\n", newtask[i].day);

        for(int j = 0; j < 3; j++){

            printf("%s\n", newtask[i].tasks[j]);

        }

    }

}

int main(){

    struct tasks newtask[7];

    strcpy(newtask[0].day ,"Monday");

    strcpy(newtask[1].day ,"Tuesday");

    strcpy(newtask[2].day ,"Wednesday");

    strcpy(newtask[3].day ,"Thursday");

    strcpy(newtask[4].day ,"Friday");

    strcpy(newtask[5].day ,"Saturday");

    strcpy(newtask[6].day ,"Sunday");

    for(int i = 0; i < 7; i++) {

        for(int j = 0; j < 3; j++) {

            newtask[i].tasks[j][0] = '\0';

        }

    }

    int cont = 1;

    char input\_day[30];

    char input\_task[40];

    while(cont == 1){

        printf("Enter the Input day: ");

        scanf("%s", input\_day);

        printf("Enter the Input Task: ");

        scanf("%s", input\_task);

        int found = 0;

        for(int i = 0; i < 7; i++) {

            if(strcmp(newtask[i].day, input\_day) == 0){

                found = 1;

                int task\_added = 0;

                for(int j = 0; j < 3; j++){

                    if(newtask[i].tasks[j][0] == '\0'){

                        strcpy(newtask[i].tasks[j], input\_task);

                        task\_added = 1;

                        break;

                    }

                }

                if(!task\_added){

                    printf("3 tasks are already entered for %s\n", newtask[i].day);

                }

                break;

            }

        }

        if(!found) {

            printf("Day not found!\n");

        }

        print\_list(newtask);

    }

    return 0;

}

OUTPUT:

PS C:\Users\Vishal\dev> gcc .\structure.c

PS C:\Users\Vishal\dev> ./a.exe

Enter the Input day: Monday

Enter the Input Task: welcme home

Monday

welcme

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

Enter the Input day: Enter the Input Task:

sdfsdf

Day not found!

Monday

welcme

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

Enter the Input day: Tuesday

Enter the Input Task: newwork

Monday

welcme

Tuesday

newwork

Wednesday

Thursday

Friday

Saturday

Sunday

1. Pointer : Even number to be before the odd numbers in the array without affecting the relative order.  
     
   Code:
2. #include <stdio.h>
3. void even\_odd(int\* arr, int size){
4. int\* end = arr + size;
5. int\* even\_pos = arr;
6. for(int\* curr = arr; curr < end; curr++) {
7. if(\*curr % 2 == 0) {
8. int even\_val = \*curr;
9. int\* shift = curr;
10. while(shift > even\_pos) {
11. \*shift = \*(shift - 1);
12. shift--;
13. }
14. \*even\_pos = even\_val;
15. even\_pos++;
16. }
17. }
18. }
19. int main(){
20. int arr[5] = {1,3,2,56,5};
21. int \*ptr = arr;
22. even\_odd(ptr,5 );
23. // Print the rearranged array
24. for(int i = 0; i < 5; i++) {
25. printf("%d ", arr[i]);
26. }
27. printf("\n");
28. return 0;
29. }

OUTPUT:

PS C:\Users\Vishal\dev> gcc .\pointer.c

PS C:\Users\Vishal\dev> ./a.exe

2 56 1 3 5

3. Arrays : 2D matrix most efficient way of finding the key.

Code:

#include <stdio.h>

#include <time.h>

int searchMatrix(int n, int matrix[n][n], int key) {

    int row = 0, col = n - 1;

    while (row < n && col >= 0) {

        if (matrix[row][col] == key)

            return 1;

        else if (matrix[row][col] > key)

            col--;

        else

            row++;

    }

    return 0;

}

int main() {

    clock\_t start, end;

    start = clock();

    int n = 4;

    int matrix[4][4] = {

        {10, 20, 30, 40},

        {15, 25, 35, 45},

        {27, 29, 37, 48},

        {32, 33, 39, 50}

    };

    int key = 29;

    if (searchMatrix(n, matrix, key))

        printf("Key %d found in matrix.\n", key);

    else

        printf("Key %d not found in matrix.\n", key);

    end = clock();

    double time\_taken = (double)(end - start) / CLOCKS\_PER\_SEC;

    printf("Time taken: %f seconds\n", time\_taken);

    return 0;

}

OUTPUT:

PS C:\Users\Vishal\dev> gcc .\keysearch.c

PS C:\Users\Vishal\dev> ./a.exe

Key 29 found in matrix.

Time taken: 0.000000 seconds