SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF DATASCIENCE AND BUSINESS SYSTEMS

18CSC161J - Fundamentals of Computer Science

STUDENT PORTFOLIO

Insert Photo

Name: DHAWAL SAHU

Register Number: RA2111042010015

Mail ID: dc5748@srmist.edu.in

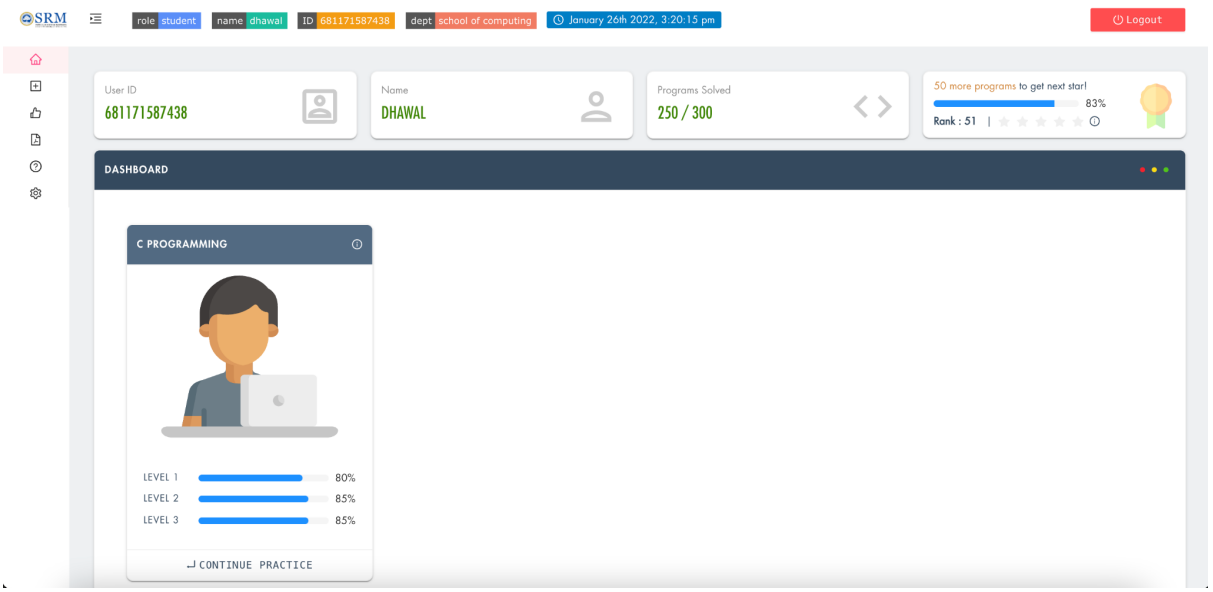
Department: Computer science

Specialization: Business System

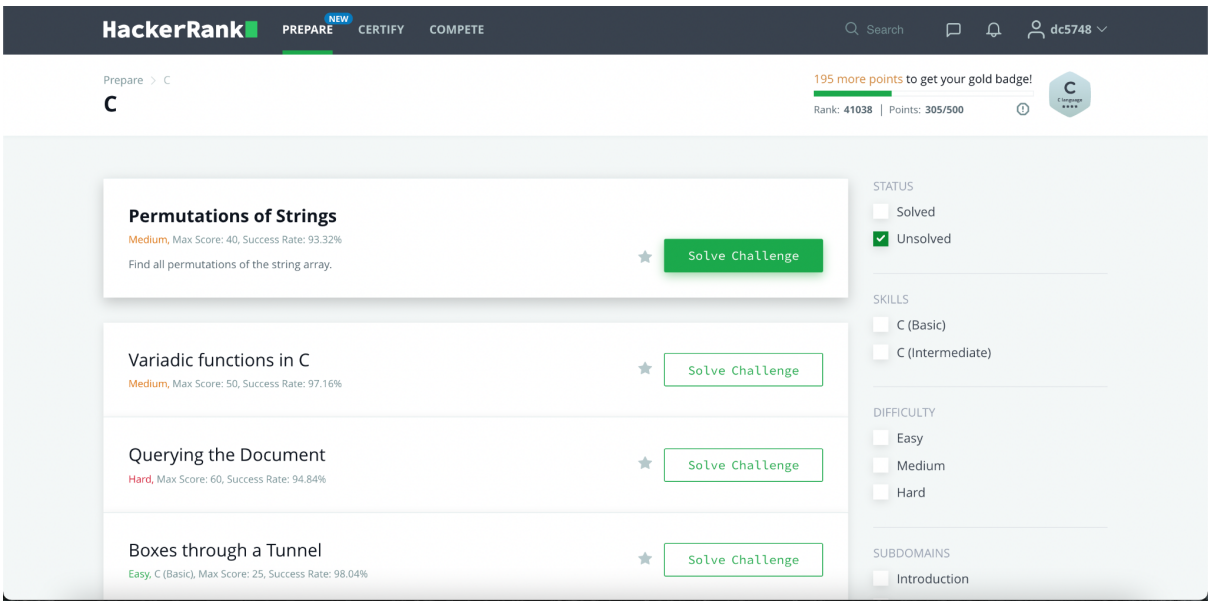
Semester: 1st

Faculty In-Charge : Dr K. SHANTHA KUMARI

ELAB DASHBOARD:



HACKERRANK DASHBOARD:



HACKERRANK QUESTIONS:

I. Sorting Array of Strings

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int lexicographic_sort(const char* a, const char* b){

    return strcmp(a, b) > 0;

}

int lexicographic_sort_reverse(const char* a, const char* b){

    return strcmp(a, b) <= 0;

}

int sort_by_number_of_distinct_characters(const char* a, const
char* b){

    int c1 = 0, c2 = 0;

    int hsh1[26] = {0}, hsh2[26] = {0};

    int n1 = strlen(a);

    int n2 = strlen(b);

   for(int i = 0; i < n1; i++){

        hsh1[a[i] - 'a'] = 1;

    }    for(int i = 0; i < n2; i++){

        hsh2[b[i] - 'a'] = 1;

    }

    for(int i = 0; i < 26; i++){

        if(hsh1[i])
```

```
            c1++;
        if(hsh2[i])
            c2++;
    }
    if( c1 != c2)
        return c1 > c2;
    else
        return strcmp(a, b)  > 0;
}
int sort_by_length(const char* a, const char* b){
    if(strlen(a) != strlen(b))
        return strlen(a) > strlen(b);
    else
        return strcmp(a, b) >  0;
}void string_sort(char** arr,const int len,int (*cmp_func)(const
char* a, const char* b)){
    for(int i = 1; i < len; i++){
        int j = i;
        char* p = arr[i];
        while(j > 0){
            if((*cmp_func)(arr[j-1],p) > 0 )
                arr[j] = arr[j-1];
            else
                break;
```

```
            j--;
        }
        arr[j] = p;
    }
}
int main()
{
    int n;
    scanf("%d", &n);
      char** arr;
    arr = (char**)malloc(n * sizeof(char*));

    for(int i = 0; i < n; i++){
        *(arr + i) = malloc(1024 * sizeof(char));
        scanf("%s", *(arr + i));
        *(arr + i) = realloc(*(arr + i), strlen(*(arr + i)) + 1);
    }
     string_sort(arr, n, lexicographic_sort);
    for(int i = 0; i < n; i++)
        printf("%s\n", arr[i]);
    printf("\n");
   string_sort(arr, n, lexicographic_sort_reverse);
    for(int i = 0; i < n; i++)
        printf("%s\n", arr[i]);
```

```
    printf("\n");
    string_sort(arr, n, sort_by_length);
    for(int i = 0; i < n; i++)
        printf("%s\n", arr[i]);
    printf("\n");
    string_sort(arr, n, sort_by_number_of_distinct_characters);
    for(int i = 0; i < n; i++)
        printf("%s\n", arr[i]);
    printf("\n");}
```

2. Dynamics Array in C

```
#include <stdio.h>
#include <stdlib.h>
int* total_number_of_books;
int** total_number_of_pages;
int main()
{
   int total_number_of_shelves;
    scanf("%d", &total_number_of_shelves);

    total_number_of_books = calloc(total_number_of_shelves,sizeof
(int));
     int total_number_of_queries;
```

```
     scanf("%d", &total_number_of_queries);
     total_number_of_pages = malloc(total_number_of_shelves *
sizeof
(int *));
     for (int i = 0; i < total_number_of_shelves; i++)
    {
        total_number_of_pages[i] = calloc(1199, sizeof(int));
    }
    while (total_number_of_queries--)
    {
        int type_of_query;
        scanf("%d", &type_of_query);

        if (type_of_query == 1)
         {
             int shelf, pages;
             scanf("%d %d", &shelf, &pages);
             total_number_of_books[shelf]++;
             int *book = total_number_of_pages[shelf];
             while (*book != 0)
             book++;
             *book = pages;
        } else if (type_of_query == 2) {
            int x, y;
```

```
            scanf("%d %d", &x, &y);
            printf("%d\n", *(*(total_number_of_pages + x) + y));
        } else {
            int x;
            scanf("%d", &x);
            printf("%d\n", *(total_number_of_books + x));
        }
    }
    if (total_number_of_books) {
        free(total_number_of_books);
    }
    for (int i = 0; i < total_number_of_shelves; i++) {
        if (*(total_number_of_pages + i)) {
            free(*(total_number_of_pages + i));
        }
    }
        if (total_number_of_pages) {
        free(total_number_of_pages);
    }
   return 0;}
```

3 Digit Frequency

```
#include <stdio.h>
#include <string.h>
#include <math.h>
```

```c
#include <stdlib.h>

int main() {

    char s[1001],i;

    int f,j;

    scanf("%s",s);

    for(i=48;i<58;i++)

    {

        f=0;

        for(j=0;j<strlen(s);j++)

        {

            if (s[j]==i)

            {

                f++;

            }

        }

        printf("%d ",f);

    }

    return 0;

}
```

4.PRINTING PATTERN USING LOOPS

```c
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```c
int main()
{

    int n;
    scanf("%d", &n);
    int len = n*2 - 1;
    for(int i=0;i<len;i++){
        for(int j=0;j<len;j++){
            int min = i < j ? i : j;
            min = min < len-i ? min : len-i-1;
            min = min < len-j-1 ? min : len-j-1;
            printf("%d ", n-min);
        }
        printf("\n");
    }
    return 0;
}
```

5.Array Reversal

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
```

```
    int num, *arr,*rev_arr, i,j;
    scanf("%d", &num);
    arr = (int*) malloc(num * sizeof(int));
    rev_arr = (int*) malloc(num * sizeof(int));
    for(i = 0; i < num; i++) {
        scanf("%d", arr + i);
    }

    for(i = num-1,j=0; i>=0&&j<num; i--,j++){
      *(rev_arr+j)=*(arr+i);
    }
    for(int i=0;i<num;i++)
    {
        *(arr+i)=*(rev_arr+i);
        printf("%d ",*(arr+i));
    }
    return 0;
}
```