

BRACT's
Vishwakarma Institute of Technology, Pune-37
Department of Electronics Engineering

Course Project Report for Data Structures & Algorithms
(ET2002)

-Guided by D.G.Kanade Sir

Course Project Name: Players Championship

Class: SY

Division: A

Semester: 1

Academic Year: 2019-20

Student Details:

| Name | GR.NO. | Roll No. |
|------------------|----------|----------|
| Ruturaj Deshmukh | 11810243 | 31 |
| Chinmay Deshmukh | 11810577 | 30 |
| Akash Gajbhiye | 11810457 | 38 |

❖ Introduction:

This is project related to system in which you can enter the scores, country, name and world rank of the player. We have designed a code where you can add, delete and display the players information. Then there are the two rounds :Round 1 and Round 2. Round 1 will eliminate half of the players according to scores and qualify them for Round 2. When you opt for the Round 2, it allows you to enter the individual scores of Round 2 of the game. Round 2 declares the top 3 winners of the game. You are also able to sort/search the players where searching is done on the basis of world rank.

❖ Concepts and Algorithms Used:

1)Stack: Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out).:

i)PUSH: void push(char[],char[],int,float) function is used to insert the new node in the stack and the arguments passed are country, name, word rank and score.

ii)POP: void pop() is the function used to pop or delete the latest inserted node in the stack.

iii)DISPLAY: void display() is the function used to display all the elements in the stack according to LIFO format.

2)Linked List: A linked list is a linear data structure where each element is a separate object. Each element (we will call it a node) of a list is comprising of two items - the data and a reference to the next node. The last node has a reference to null.

3)Linear Search: Linear search is a very simple search algorithm. In this type of search, a sequential search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection

i)Search: struct player *rank(struct player *top) is the function used in which top of the stack is passed as an argument and the search is carried out according to world rank using LINEAR SEARCH algorithm whose time complexity is $O(N)$.

4)Selection Sort: The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array:

i) The subarray which is already sorted.

ii) Remaining subarray which is unsorted

a)struct player *sort(struct player *top) is the function used to sort the players in ascending order according to their scores using SELECTION SORT algorithm whose time complexity is $O(n^2)$ and the top of the stack is sent as argument to sort all the elements in the stack.

b)struct player *sortd(struct player *top) is the function used to sort the players in ascending order according to their scores using SELECTION SORT algorithm whose

time complexity is $O(n^2)$ and the top of the stack is sent as argument to sort all the elements in the stack.

5)ROUND 1: void round1(int) which takes the count of the stack as an argument and eliminates(pop) the half of the players according to their scores from the stack.

6)ROUND 2: struct player *round2(struct player *top) is the function used which takes top of the stack as an argument and accepts the round 2 scores of players which have passed the round 1 and pass it to sortd function which sorts it in descending order according to scores and declares the top 3 winners of the game.

❖ Code:

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
#include<stdlib.h>
#include<malloc.h>
```

```
struct player
{
    char country[10];
    char name[10];
    float score;
    int rank;
    struct player *next;
}*top,*top1,*temp,*tem,*top2;
```

```
void push(char[],char[],int,float);
void pop();
void display();
void destroy();//Destroys all Stack elements
void round1(int);//Round 1 eliminates half of the Players
struct player *round2(struct player *top);//Round 2 Declars the top 3 Winners
struct player *sort(struct player *top);//Sorting in Ascending order
struct player *sortd(struct player *top);//In Descending
struct player *rank(struct player *top);//Linear Search By Rank
```

```
int main()
{
    //clrscr();
    int ch;
    int cnt=0;
    char cntry[10];
    float scr;
    char nam[10];
```

```

int rnk;

top=NULL;

while (1)
{
    printf("\n\t\t\t\t\t\t\t1) Add a Player   5) Round1 \n\t\t\t\t\t\t\t2)
Delete a Player   6) Round2 ");
    printf("\n\t\t\t\t\t\t\t3) Display 7) Destroy \n\t\t\t\t\t\t\t4)
Sort/Search8) Exit");

    printf("\n\n Enter choice : ");
    scanf("%d", &ch);

    switch (ch)
    {
        case 1:
            int b,g;
            printf("Add Minimum 6 Players Details");
            printf("\nHow many Player are Playing:");
            scanf("%d",&b);

            for(int c=0;c<b;c++)
            {
                printf("\nName: ");
                scanf("%s",nam);
                printf("Country: ");
                scanf("%s",cntry);
                printf("Ranking in World: ");
                scanf("%d",&rnk);
                printf("Score: ");
                scanf("%f",&scr);
                push(nam,cntry,rnk,scr);
                cnt++;
            }
            break;
        case 2:
            pop();
            break;
        case 8:
            exit(0);
        case 3:
            display();
    }
}

```

```

break;
case 7:
destroy();
break;
case 4:
int p;
printf("\n1) Sort\n2) Search by WorldRank\n");
scanf("%d",&p);
switch(p)
{
case 1:
int k;
printf("\n1) Ascending\n2) Descending\n");
scanf("%d",&k);
switch(k)
{
case 1:
top=sort(top);
display();
break;
case 2:
top=sortd(top);
display();
break;
}
break;
case 2:
top=rank(top);
break;
}
break;
case 5:
top=sort(top);
display();
printf("\n-----\n");
printf("Name  Country  Status ");
printf("\n-----\n");

round1(cnt);
break;
case 6:

top=round2(top);
break;
default :

```

```

                printf(" Wrong choice, Please enter correct choice ");
                break;
            }
        }
    }
}

```

```

void push(char nam[],char cntry[],int rnk,float scr)
{
    if (top == NULL)
    {
        top=(struct player*)malloc(sizeof(struct player));
        top->next = NULL;
        strcpy(top->name,nam);
        strcpy(top->country,cntry);
        top->rank=rnk;
        top->score=scr;
    }
    else
    {
        temp=(struct player*)malloc(sizeof(struct player));
        strcpy(temp->name,nam);
        strcpy(temp->country,cntry);
        temp->rank=rnk;
        temp->score=scr;
        temp->next = top;
        top = temp;
    }
}

```

```

/* Display stack elements */
void display()
{
    top1 = top;
    printf("\n-----\n");
    printf("Name  Country  WorldRank  Score");
    printf("\n-----\n");

    if (top1 == NULL)
    {
        printf("Players list is empty");
        return;
    }

    while (top1 != NULL)
    {

```

```

        printf("\n%s\t%s\t %d\t    %f", top1->name, top1->country, top1-
>rank, top1->score);
        top1 = top1->next;
    }
}

```

/* Pop Operation on stack */

```

void pop()
{
    top1 = top;
    if (top1 == NULL)
    {
        printf("\n Error : Trying to delete from empty list");
        return;
    }
    else
        top1 = top1->next;
    printf("\n%s    %s    Eliminated", top->name, top->country);
    free(top);
    top = top1;
}

```

```

void destroy()
{
    top1 = top;
    while (top1 != NULL)
    {
        top1 = top->next;
        free(top);
        top = top1;
        top1 = top1->next;
    }
    free(top1);
    top = NULL;
    printf("\n All players deleted");
}

```

```

struct player *sort(struct player *top)
{
    if(top==NULL)
    {
        printf("No player, therefore cannot sort");
        return 0;
    }
}

```

```

    }
    struct player *p,*q,*r,*s,*tmp;
    for(r=p=top;p->next!=NULL;r=p,p=p->next)
    {
        for(s=q=p->next;q!=NULL;s=q,q=q->next)
        {
            if(p->score > q->score)
            {
                tmp=p->next;
                p->next=q->next;
                (q->next)=tmp;

                if(p!=top)
                    r->next=q;
                s->next=p;

                if(p==top)
                    top=q;
                tmp=p;
                p=q;
                q=tmp;
            }
        }
    }

    return top;
}

```

```

struct player *sortd(struct player *top)
{
    if(top==NULL)
    {
        printf("No player, therefore cannot sort");
        return 0;
    }
    struct player *p,*q,*r,*s,*tmp;
    for(r=p=top;p->next!=NULL;r=p,p=p->next)
    {
        for(s=q=p->next;q!=NULL;s=q,q=q->next)
        {
            if(p->score < q->score)
            {
                tmp=p->next;
                p->next=q->next;

```



```

        (q->next)=tmp;

        if(p!=top)
            r->next=q;
        s->next=p;

        if(p==top)
            top=q;
        tmp=p;
        p=q;
        q=tmp;
    }
}

return top;
}

```

```

struct player *rank(struct player *top)
{
    if(top==NULL)
    {
        printf("\n\nRecord Not Found Cannot Search");
        return 0;
    }
    struct player *p,*temp;;
    int search,cntr=0;
    p=top;
    printf("Enter World Rank: ");
    scanf("%d",&search);
    while(p!=NULL)
    {
        if(p->rank==search)
        {
            cntr++;
            break;
        }
        p=p->next;
    }
    if(cntr==1)
        printf("\n%s has World Rank %d\n\n",p->name,search);
    else if(cntr==0)
        printf("\nNone of the Player has %d as his WorldRank\n\n",search);
    return top;
}

```

```
}
```

```
void round1(int cnt)
{
    int n=cnt;
    for(int i=1;i<=(n/2);i++)
    {
        pop();
    }
}
```

```
struct player *round2(struct player *top)
{
    float scor;
    top1 = top;
    top2=top;
    if (top1 == NULL)
    {
        printf("Players list is empty");
        return 0;
    }
    printf("\n-----\n");
    printf("Name      2nd Round Score");
    printf("\n-----\n");
```

```
    while (top1 != NULL)
    {
        printf("%s\t:  ", top->name);
        //printf("2nd Round Score:");
        scanf("%f",&scor);
        top->score=scor;
        top1 = top1->next;
        top =top->next;
    }
    top=sortd(top2);
```

```
    printf("\n-----\n");
    printf("Name Country    Medal");
```

```
printf("\n-----\n");
```

```
for(int i=1;i<=3;i++)
{
    top2 = top;
    if (top2 == NULL)
    {
        printf("\n Error : Trying to delete from empty list");
        return 0;
    }
    else
        top2 = top2->next;

    if(i==1)
        printf("\n %s    %s    Gold Medal", top->name,top->country);
    if(i==2)
        printf("\n %s    %s    Silver Medal", top->name,top-
>country);
    if(i==3)
        printf("\n %s    %s    Bronze Medal", top->name,top-
>country);
    free(top);
    top = top2;
}

return top;

}
```

❖ Observations:

1)Round 1 Output:

```
Enter choice : 5
```

| Name | Country | WorldRank | Score |
|-----------|-----------|-----------|------------|
| Akash | Australia | 54 | 56.000000 |
| Ruturaj | India | 12 | 234.000000 |
| Ashitosh | India | 14 | 345.000000 |
| Vaibhav | England | 67 | 588.000000 |
| Chinmay | India | 78 | 990.000000 |
| Prathmesh | India | 56 | 999.000000 |

| Name | Country | Status |
|----------|-----------|------------|
| Akash | Australia | Eliminated |
| Ruturaj | India | Eliminated |
| Ashitosh | India | Eliminated |

2)Round 2 Output:

```
Enter choice : 6
```

| Name | 2nd Round Score |
|-----------|-----------------|
| Prathmesh | : 333 |
| Chinmay | : 123 |
| Vaibhav | : 789 |

| Name | Country | Medal |
|-----------|---------|--------------|
| Vaibhav | England | Gold Medal |
| Prathmesh | India | Silver Medal |
| Chinmay | India | Bronze Medal |

3)Sort/Search:

```
Enter choice : 4
1) Sort
2) Search by WorldRank
1
1) Ascending
2) Descending
2
-----
Name      Country   WorldRank   Score
-----
Prathmesh      India      56          999.000000
Chinmay India      78          990.000000
Vaibhav England    67          588.000000
```

4)Display:

```
Enter choice : 3
-----
Name      Country   WorldRank   Score
-----
Vaibhav England    67          588.000000
Chinmay India      78          990.000000
Prathmesh      India      56          999.000000
```

❖ Conclusions and Future Scope:

- 1) It is fully fledged program related to application of sorting,display,searching of players in a league.
- 2) It can further be used in various applications and quizzes where rounds matter for any competition

- 3) In future it can be implemented using Circular Queue and tree structures like BST and it is our first step towards implementation of stack and data structures in real life applications.