A1 Reverse a string

#include <stdio.h>

#include <string.h>

#define SIZE 20

void reverseString(char []);

int main()

{

char input[SIZE];

scanf("%s",input);

reverseString(input);

printf("Reversed string: %s",input);

return 0;

}

void reverseString(char string[])

{

int length=strlen(string);

int i=0;

int mid=0;

if(length%2==0)

mid=length/2;

else

mid=(length/2)+1;

while(i<mid)

{

char temp=string[length-i-1];

string[length-i-1]=string[i];

string[i]=temp;

i++;

}

}

A2

Reverse string with given intervals

#include <stdio.h>

#include <string.h>

#define CAP 20

void rotate\_string(char [],int,int);

void reverseString(char [],int,int);

int main()

{

char str[CAP];

int key,len;

printf("Enter the string: ");

scanf("%s",str);

printf("Enter the key: ");

scanf("%d",&key);

len = strlen(str);

rotate\_string(str,len,key);

printf("%s",str);

}

void rotate\_string(char str[],int n,int key)

{

int i;

int end;

for(i=0;i<n;)

{

if(i+key-1<n)

end=i+key-1;

else

end=n-1;

reverseString(str,i,end);

i+=key;

}

}

void reverseString(char str[],int start,int end)

{

char temp;

while(start<end)

{

temp=str[start];

str[start]=str[end];

str[end]=temp;

start++;

end--;

}

}

A3 WAP to which should have the following functionalities

1. scan employee recs

2. display

3. search for a particulary employee based on

a. id

b. lName

4. search for employess whos salary fall in

between 6-10K and display only those records

5. update employee rec to change his/her lName

#include <stdio.h>

#include <string.h>

#define CAP 20

struct Employee

{

int Id;

float Salary;

char Title[2];

char fName[CAP];

char lName[CAP];

};

typedef struct Employee EMP;

void scanEmployees(EMP [],int);

void displayEmployees(EMP [],int);

void searchById(EMP [],int,int);

void displayRequired(EMP [],int);

void searchBylName(EMP [],int,char []);

void searchBySalary(EMP [],int,float);

void updateEmployeeName(EMP [],int,int);

int main()

{

int size;

int choice,id;

float salary;

char lname[CAP];

printf("Number of emplyees: ");

scanf("%d",&size);

EMP e[size];

while(1)

{

printf("\nMenu:");

printf("\n1.Scan Employees");

printf("\n2.Display Employees details");

printf("\n3.Search");

printf("\n4.Update Employee lname");

printf("Enter the choice: ");

scanf("%d",&choice);

switch(choice)

{

case 1:

scanEmployees(e,size);

break;

case 2:

displayEmployees(e,size);

break;

case 3:

printf("\nSearch options:");

printf("\n1.Search by ID");

printf("\n2.Search by lName");

printf("\n3.Search by salary");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("Enter ID: ");

scanf("%d",&id);

searchById(e,size,id);

break;

case 2:

printf("Enter lName: ");

scanf("%c",lname);

searchBylName(e,size,lname);

break;

case 3:

printf("Enter salary: ");

scanf("%f",&salary);

searchBySalary(e,size,salary);

break;

}

break;

case 4:

printf("Enter the id: ");

scanf("%d",&id);

updateEmployeeName(e,size,id);

break;

case 5:

printf("Exited");

flag=1;

break;

default:

printf("Enter the valid choice");

break;

}

if(flag==1) break;

}

return 0;

}

void scanEmployees(EMP e[],int n)

{

for(int i=0;i<n;i++)

{

printf("\nEnter employee%d details: \n",i+1);

printf("\nID: ");

scanf("%d",&e[i].Id);

printf("\nName: ");

scanf("%s ",e[i].Title);

scanf("%s ",e[i].fName);

scanf("%s",e[i].lName);

printf("\nSalary: ");

scanf("%f",&e[i].Salary);

printf("\n");

}

}

void displayEmployees(EMP e[],int n)

{

printf("\nDetails of employees are: \n");

for(int i=0;i<n;i++)

{

printf("\nEmployee %d",i+1);

printf("\nID: %d",e[i].Id);

printf("\nName: %s %s %s ",e[i].Title,e[i].fName,e[i].lName);

printf("\nSalary: %f",e[i].Salary);

printf("\n");

}

}

void searchById(EMP e[],int n,int id)

{

int j=0;

for(int i=0;i<n;i++)

{

if(e[i].Id==id)

{

displayRequired(e,i);

j++;

break;

}

}

if(j==0)

printf("Invalid Id");

}

void searchBylName(EMP e[],int n,char lname[])

{

int j=0;

for(int i=0;i<n;i++)

{

if(strcmp(e[i].lName,lname)==0)

{

displayRequired(e,i);

j++;

break;

}

if(j==0)

printf("Enter valid lname");

}

}

void displayRequired(EMP e[],int i)

{

printf("\nID: %d",e[i].Id);

printf("\nName: %s %s %s",e[i].Title,e[i].fName,e[i].lName);

printf("\nSalary: %f",e[i].Salary);

printf("\n");

}

void searchBySalary(EMP e[],int n,float salary)

{

int j=0;

for(int i=0;i<n;i++)

{

if(e[i].Salary>6000 && e[i].Salary<10000)

{

displayRequired(e,i);

j++;

break;

}

}

if(j==0)

printf("Not found employee between given salary range");

}

void updateEmployeeName(EMP e[],int n,int id)

{

for(int i=0;i<n;i++)

{

char lname[CAP];

if(e[i].Id==id)

{

printf("Enter the lname: ");

scanf("%s",lname);

strcpy(e[i].lName,lname);

printf("Updated last name to %s for id %d",lname,id);

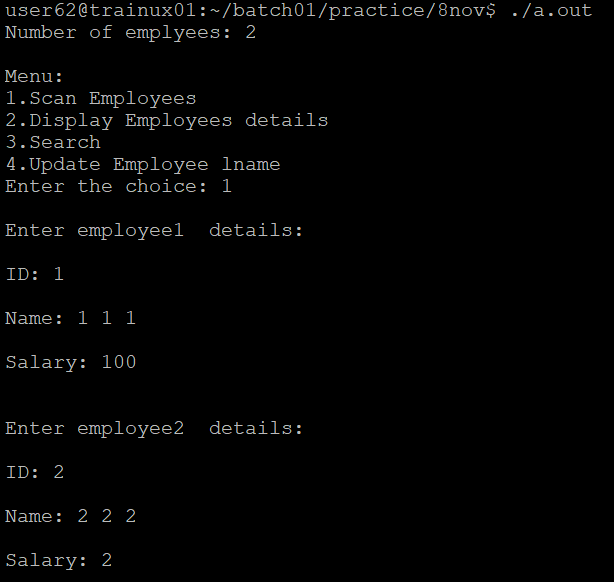
break;

}

}

}

Output:



A screenshot of a computer

Description automatically generated

