1. Write a program to read text lines from the interface. Use user defined functions and display the longest line read. Do not use any inbuilt function.

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int lengs(char a[]);

void main()

{

char str[100][100];

int i;

int L;

int j;

int b;

int n;

printf(" enter no. of lines: ");

scanf("%d",&n);

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

{

gets(str[i]);

}

b=lengs(str[0]);

L=b;

for(i=1;i<n;i++)

{

b=lengs(str[i]);

if(L<b)

{

L=b;

j=i;

}

}

printf(" The longest line is: ");

puts(str[j]);

return;

}

int lengs(char a[100])

{

int i;

for(i=0;a[i]!='\0';i++);

return i;

}

1. Rewrite the above program to find a pattern of letters in the set of lines read. Use user defined functions and display the lines having the matched pattern of letters. Do not use any inbuilt function.

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int lengs(char a[]);

void main()

{

char str[100][100];

int k[100];

char strd[100];

int i;

int L;

int j;

int b;

for(i=0;i<10;i++)

{

gets(str[i]);

}

printf(" enter pattern: ");

scanf("%s",strd);

i=0;

for(i=0;i<10;i++)

{

j=0;

b=0;

while(str[i][j]!='\0')

{

if(str[i][j]!=strd[b])

{

j++;

}

else

{

j++;

b++;

while(strd[b]!='\0')

{

if(str[i][j]!=strd[b])

{

j++;

b=0;

break;

}

else

{

j++;

b++;

}

}

if(strd[b]=='\0')

{

puts(str[i]);

}

}

}

}

}

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

void main()

{

FILE \*fp;

fp=fopen("mai.txt","r");

char str[100][100];

char strd[100];

char strs[100];

int i;

int j;

int b;

int n;

int m,h,g;

printf(" enter no. of lines: ");

scanf("%d",&n);

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

{

for(j=0;str[i][j]!='\0';j++)

{

fscanf(fp,"%c",&str[i][j]);

printf("%c",str[i][j]);

}

}

printf(" enter pattern to search: ");

scanf("%s",strd);

printf(" enter patter to rewrite: ");

scanf("%s",strs);

i=0;

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

{

j=0;

b=0;

while(str[i][j]!='\0')

{

if(str[i][j]!=strd[b])

{

j++;

}

else

{

j++;

b++;

while(strd[b]!='\0')

{

if(str[i][j]!=strd[b])

{

j++;

b=0;

break;

}

else

{

j++;

b++;

}

}

if(strd[b]=='\0')

{

for(int n=0,m=0;strs[m]!='\0';b--,m++)

{

n++;

int g=j-b;

str[i][g]=strs[m];

}

puts(str[i]);

fprintf(fp,"%s",str[i]);

}

}

}

}

fclose(fp);

}

#include <stdio.h>

#include <stdlib.h>

int main()

{

int a[25];

int b[25];

int i,j;

int n=5;

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

{

scanf("%d",&a[i]);

}

printf("the elements are:\n");

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

{

printf("%d\t",a[i]);

}

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

{

int m=0;

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

{

if(a[i]==a[j])

{

m++;

}

}

b[i]=m;

printf(" \n%d is repeated %d times\n",a[i],m);

}

int l;

l=b[0];

j=0;

for(i=1;i<n;i++)

{

if(b[i]>l)

{

j=i;

}

}

printf(" %d is the mode",a[j]);

}

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define STACKSIZE 10000

struct stack

{

int s[STACKSIZE];

int top;

};

int stacktop(struct stack stk )

{

printf("the top value is",stk.s[stk.top]);

}

int full(struct stack stk)

{

if(stk.top==STACKSIZE-1)

return 1;

else

return 0;

}

int empty(struct stack stk)

{

if(stk.top==-1)

return 1;

else

return 0;

}

void push(struct stack \*ps,int data)

{

ps->top=ps->top+1;

ps->s[ps->top]=data;

}

int pop(struct stack \*ps)

{

int data=ps->s[ps->top];

ps->top=ps->top-1;

return data;

}

void display(struct stack stk)

{

for(int i=stk.top;i>=0;i--)

{

printf("%d\n",stk.s[i]);

}

}

int main()

{

int a[STACKSIZE];

int i,j;

FILE \*fp1,\*fp2,\*fp3,\*fp4,\*fp5;

fp1=fopen("inp.txt","a");

fp2=fopen("stack.txt","a");

fp3=fopen("push.txt","a");

fp4=fopen("pop.txt","a");

fp5=fopen("opr.txt","a");

if(fp1==NULL)

{

printf("file not created");

exit(0);

}

if(fp2==NULL)

{

printf("file not created");

exit(0);

}

if(fp3==NULL)

{

printf("file not created");

exit(0);

}

if(fp4==NULL)

{

printf("file not created");

exit(0);

}

if(fp5==NULL)

{

printf("file not created");

exit(0);

}

int low,up,n;

printf(" enter the lower and upper range:\n ");

scanf("%d%d",&low,&up);

printf(" enter the n value: ");

scanf("%d",&n);

srand(time(NULL));

printf(" random no.s are:\n");

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

{

a[i]=(rand()%(up-low+1))+low;

printf("%d\t",a[i]);

fprintf(fp1,"%d\t",a[i]);

}

printf("\n");

fclose(fp1);

struct stack stk;

int choice,data;

stk.top=-1;

i=0;

do

{

printf("1-> push \t 2-> pop \t 3-> stacktop \t 4-> display \t 5->exit\n");

scanf("%d",&choice);

switch(choice)

{

case 1:if(full(stk))

printf("\n stack overflow\n");

else

{

push(&stk,a[i]);

fprintf(fp3,"%d\n",a[i]);

fprintf(fp5,"push of %d\n",a[i]);

i++;

printf("\n");

}

break;

case 2:if(empty(stk))

printf("\n stack underflow\n");

else

{

data=pop(&stk);

printf("popped data is %d \n",data);

fprintf(fp4,"%d\n",data);

fprintf(fp5,"pop of %d\n",data);

printf("\n");

}

break;

case 3:if(empty(stk))

printf(" \n stack empty \n");

else

printf(" \n stack top data is %d \n",stacktop(stk));

break;

case 4:if(empty(stk))

printf(" \n stack element empty \n");

else

{

printf(" \n stack elements \n");

display(stk);

}

break;

case 5:break;

}

}while(choice!=5);

for(int i=stk.top;i>=0;i--)

{

fprintf(fp2,"%d\n",stk.s[i]);

}

fclose(fp2);

fclose(fp3);

fclose(fp4);

fclose(fp5);

return 0;

}

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define SIZE 10000

float bubble(int a[],int n);

float select(int a[],int n);

int main()

{

int a[SIZE],s[SIZE];

int i,j,temp;

float dur,p,q;

clock\_t start,end;

FILE \*fp,\*fp1,\*fp2;

fp=fopen("inp.txt","w");

fp1=fopen("output.txt","w");

fp2=fopen("log.txt","w");

if(fp==NULL)

{

printf("file not created");

exit(0);

}

if(fp1==NULL)

{

printf("file not created");

exit(0);

}

if(fp2==NULL)

{

printf("file not created");

exit(0);

}

int low,up,n;

printf(" enter the lower and upper range:\n ");

scanf("%d%d",&low,&up);

printf(" enter the n value: ");

scanf("%d",&n);

srand(time(NULL));

printf(" random no.s are:\n");

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

{

a[i]=(rand()%(up-low+1))+low;

printf("%d\t",a[i]);

fprintf(fp,"%d\t",a[i]);

}

printf("\n");

fclose(fp);

q=bubble(a,n);

fprintf(fp2,"bubble sort\nvalue of x:%d,time required: %f",n,q);

printf(" sorted data\n");

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

{

printf("%d\t",a[i]);

fprintf(fp1,"%d\t",a[i]);

}

p=select(a,n);

fprintf(fp2,"select sort\nvalue of x:%d,time required: %f",n,p);

fclose(fp);

fclose(fp1);

fclose(fp2);

return 0;

}

float bubble(int a[SIZE],int n)

{

clock\_t start,end;

int i,j,temp;

start=clock();

float dur;

for(i=1;i<n;i++)

{

for(j=0;j<(n-i);j++)

{

if(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

end=clock();

dur=(float)(end-start);

printf(" duration for bubble sort is: %f",dur);

return dur;

}

float select(int a[],int n)

{

clock\_t start,end;

int i,j,temp,s;

start=clock();

float dur;

for(i=0;i<n-1;i++)

{

s=i;

for(j=i+1;j<n;j++)

{

if(a[j]<a[s])

{

s=j;

}

}

if(i!=s)

{

temp=a[i];

a[i]=a[s];

a[s]=temp;

}

}

end=clock();

dur=(float)(end-start);

printf(" duration for select sort is: %f",dur);

return dur;

}