Task One

Write a function whose prototype is void exchange(int \*p, int \*q); that takes two

pointers to integer variables and exchanges the values in those variables.

#include<stdio.h>

void exchange(int \*p, int \*q){//function to exchange two variable

int temp;

temp=\*p;

\*p=\*q;

\*q= temp;

}

int main(){

int num1, num2;

printf("enter two numbers\n");

scanf("%d%d", &num1, &num2);

printf("\nThe number before exchange: %d %d", num1, num2);

exchange(&num1, &num2);

printf("\nThe number after exchange: %d %d", num1, num2);

return 0;

}

Task Two:

We have two arrays A and B, each containing 10 integers. Write a function that checks if every

element of array A is equal to its corresponding element in array B. In other words, the function

must check if A[0] is equal to B[0] and A[1] is equal to B[1] and so on. The function must accept

only pointer values and return a Boolean true for equal and false for unequal.

#include<stdio.h>

int array(int \*p,int \*q);

int main(){

int \*p,\*q,i;

int A[10],B[10];

array(&A[i],&B[i]);

return 0;

}

int array(int \*p,int \*q){

int A[10];

int B[10];

int i;

p=&A[i];

q=&B[i];

printf("enter the first array elements\n");

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

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

}

printf("enter the second array elements \n");

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

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

}

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

if(p[i]==q[i]){

printf("true\n");

}

else{

printf("false\n");}

}

}

Task Three

In this activity, you are required to create a simple program that stores the student records. For

each student, you have to keep their registration number, first name, second name, sex, age, year

of study and degree program. Note that every student’s registration is unique.

Your program must contain a menu that supports the following operations

(i) Adding new records at a start of a list, at the middle and at the end of the list

(ii) Modify a record

(iii) Delete a record

(iv) Print all elements in the list

(v) Display a particular student's details when his/her registration number is provided

(vi) Quit from an application when the user press ‘q’ or ‘Q’ on the keyboard

The implementation of this program should be based on the linked list ADT.

Validation:

(a) The program must avoid registration number collision i.e. the program only accepts

records which don’t have registration number collision

(b) The program should reject a new record with an empty registration number (c) Before

deleting a record a user has to confirm first

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct studentInfo{

char regno[15];

char fname[15];

char secname[15];

char sex[6];

int age;

int yearofStudy;

char dProgram[25];

struct studentInfo \*link;

}STUDENT;

STUDENT \*head=NULL;

//function declaration

void homePage();

void addRecord();

void addFirstEnd();

void addMiddle();

void modifyRecord();

void deleteRecord();

void printAll();

void printOnce();

//main function

int main(){

//call home page function

homePage();

system("pause");

}

//homepage function

void homePage(){

printf("\n\nSTUDENT INFORMATION SYSTEM (SIS)\n\n");

printf("1. Add new record at the start of a list.\n");

printf("2. Add new record at the end of a list.\n");

printf("3. Add new record at the middle of a list.\n");

printf("4. Modify a record.\n");

printf("5. Delete a record.\n");

printf("6. Print all records.\n");

printf("7. Print particular record.\n");

printf("choose q or Q to quit a program.\n\n");

printf("Enter your choice: ");

char choice;

scanf("%s",&choice);

switch(choice){

case '1':

case '2':addFirstEnd();break;

case '3':addMiddle();break;

case '4':modifyRecord();break;

case '5':deleteRecord();break;

case '6':printAll();break;

case '7':printOnce();break;

case 'q':

case 'Q':printf("Enjoy the system.\n\n");exit(0);break;

default:printf("Invalid choice, please enter again.\n\n");

homePage();

break;

}

}

//add first

void addFirstEnd(){

STUDENT \*stds;

if(head==NULL){

stds=(STUDENT \*)malloc(sizeof(STUDENT));

stds->link=head;

head=stds;

}

else{

STUDENT \*current=head;

stds=(STUDENT \*)malloc(sizeof(STUDENT));

while(current->link!=NULL){current=current->link;}

current->link=stds;

stds->link=NULL;

}

printf("\nProvide student information:\n\n");

printf("Registration number: "); scanf("%s",&stds->regno);

STUDENT \*check=head;

while(check->link!=NULL){

if(strcmp(check->regno,stds->regno)==0){

printf("Registration number already exist.\n\n");

free(stds);

addFirstEnd();

break;

}

check=check->link;

}

printf("First name: "); scanf("%s",&stds->fname);

printf("Second name: "); scanf("%s",&stds->secname);

printf("sex: "); scanf("%s",&stds->sex);

printf("Age: "); scanf("%d",&stds->age);

printf("Year of study: "); scanf("%d",&stds->yearofStudy);

printf("Degree program: "); scanf("%s",&stds->dProgram);

printf ( "\nStudent record successfully inserted.\n" );

homePage();

}

//add at the middle

void addMiddle(){

if(head==NULL){

printf("List is empty.");

homePage();

}

else {

char regnum[15];

printf("\n\nEnter registration number that new student will be inserted after: ");

scanf("%s",&regnum);

STUDENT \*current=head;

STUDENT \*std=(STUDENT \*)malloc(sizeof(STUDENT));

while(current!=NULL){

if(strcmp(current->regno,regnum)==0){

printf("\nProvide student information:\n\n");

printf("Registration number: "); scanf("%s",&std->regno);

STUDENT \*check2=head;

while(check2!=NULL){

if(strcmp(check2->regno,std->regno)==0){

printf("Registration number already exist.\n\n");

free(std);

homePage();

break;

}

check2=check2->link;

}

printf("First name: "); scanf("%s",&std->fname);

printf("Second name: "); scanf("%s",&std->secname);

printf("sex: "); scanf("%s",&std->sex);

printf("Age: "); scanf("%d",&std->age);

printf("Year of study: "); scanf("%d",&std->yearofStudy);

printf("Degree program: "); scanf("%s",&std->dProgram);

std->link=current->link;

current->link=std;

printf ( "\nStudent record successfully inserted.\n" );

homePage();

break;

}

current=current->link;

}

}

}

//function to modify records

void modifyRecord(){

char regNum[15];

printf("Enter registration number you want to modify: ");

scanf("%s",&regNum);

STUDENT \*temp= head ;

while(temp!=NULL)

{

if ( strcmp(temp->regno,regNum)==0 )

{

printf("First name: "); scanf("%s",&temp->fname);

printf("Second name: "); scanf("%s",&temp->secname);

printf("sex: "); scanf("%s",&temp->sex);

printf("Age: "); scanf("%d",&temp->age);

printf("Year of study: "); scanf("%d",&temp->yearofStudy);

printf("Degree program: "); scanf("%s",&temp->dProgram);

homePage();

return ;

}

else{

printf ( "\nRegistration number %s is not in the list\n", regNum ) ;

modifyRecord();

}

}

}

//function to delete records

void deleteRecord(){

char num[15];

char dltcmf;

printf("Are you sure you want to delete?\n");

printf("press Y to delete, N to cancel.\n");

scanf("%s",&dltcmf);

if(dltcmf=='Y' || dltcmf=='y'){

printf("Enter registration number you want to delete: ");

scanf("%s",&num);

STUDENT \*old, \*temp ;

temp = head ;

while ( temp != NULL ){

if ( strcmp(temp->regno,num)==0 ){

if ( temp == head )

head = temp -> link ;

else

old -> link = temp -> link ;

free ( temp ) ;

printf ( "\nStudent record successfully deleted.\n" );

homePage();

return ;

}

else{

old = temp ;

temp = temp -> link ;

}

}

printf ( "\nStudent record not found.\n" ) ;

homePage();

}

else if(dltcmf=='N' || dltcmf=='n'){

homePage();

}

else{

printf ( "\nWrong selection\n" ) ;

deleteRecord();

}

}

//print all records

void printAll(){

if(head==NULL){

printf("List is empty.");

homePage();

}

else{

STUDENT \*load =head;

printf("\n\nStudent information:\n\n");

while(load!=NULL){

printf("Registration number\t: %s\n",load->regno);

printf("Full name\t\t: %s %s\n",load->fname,load->secname);

printf("sex\t\t\t: %s\n",load->sex);

printf("Age\t\t\t: %d\n",load->age);

printf("Year of study\t\t: %d\n",load->yearofStudy);

printf("Degree program\t\t: %s\n\n\n",load->dProgram);

load=load->link;

}

homePage();

}

}

//print particular record

void printOnce(){

if(head==NULL){

printf("List is empty.");

homePage();

}

else{

char reg[15];

printf("Enter registration number: ");

scanf("%s",&reg);

STUDENT \*load =head;

printf("\n\nStudent information:\n\n");

while(load!=NULL){

if(strcmp(load->regno,reg)==0){

printf("Registration number\t: %s\n",load->regno);

printf("Full name\t\t: %s %s\n",load->fname,load->secname);

printf("sex\t\t\t: %s\n",load->sex);

printf("Age\t\t\t: %d\n",load->age);

printf("Year of study\t\t: %d\n",load->yearofStudy);

printf("Degree program\t\t: %s\n\n\n",load->dProgram);

homePage();

break;

}

load=load->link;

}

printf("Student record not found.\n");

homePage();

}

}

Task Four

Write a program that will compute statistics (average, maximum, minimum, and standard

deviation) for a set of numbers. Ask the user for the number of numbers to process and read the

numbers from standard input (stdin). The program should then display the numbers entered and

the statistics, all nicely formatted. What data structures did you use and why?

#include <math.h>

int main () {

float average, SD, sum=0.0;

int max, min, i, size, number[size], count=0;

printf("Entert size of number\n");

scanf("%d", & size);

printf("Enter your numbers\n");

for(i=0;i<size;i++){

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

sum= sum +number[i];

count++;

}

average =sum/size;

max = number[0];

min = number[0];

SD = sqrt(sum/size);

for(i=0;i<size;i++){

if(number[i]>max){

max = number[i];

}

}

for(i=0;i<size;i++){

if (number[i]<min){

min = number[i];

}

}

printf(" the average is %f\ n",average);

printf("the max number is %d\n",max);

printf("the min number is %d\n",min);

printf("the Standard deviation is %f" , SD);

return 0;

}

Task five:

#include<stdio.h>

#include<math.h>

int main(){

float avg, std, var, sum1=0, k;

int size, num, I, max, min, sum = 0, j=0;

printf(“\nEnter number of element: “);

scanf(“%d”, &size); int stat[size];

//alocate element to array

for(I = 0; i<size; i++)

{

printf(“\nEnter element %d: “, i+1);

scanf(“%d”, &stat[i]);

sum = sum + stat[i];

}

/\* display array entered \*/

for(i=0; i<size; i++) {

printf(“%d\t”, stat[i]);

j++;

if(j==4) {

printf(“\n”);

j=0;

}

} //maximum and minimum number

int temp;

for(i= 0; i<size; i++) {

for(j=i+1; j<size; j++) {

if(stat[i] < stat[j]) {

temp = stat[j];

stat[j] = stat[i];

stat[i] = temp; } } }

max = stat[0];

min = stat[size-1];

avg=sum/size;

for(i=0; i<size; i++) {

k=pow((stat[i]-avg),2);

sum1 = sum1 + k;

} var = sum1/size;

std = sqrt(var);

printf(“\nstd = %.2f”, std);

printf(“\nvar =%.2f”, var);

printf(“\nsum = %d”, sum);

printf(“\nmax = %d”, max);

printf(“\nmin = %d”, min);

printf(“\navg = %.2f”, avg);

Return 0;

}

Task six:

/\* A program to Compute Statistics for a set of numbers(average, maximum, minimum and Standard deviation)\*/

#include<stdio.h>

#include<math.h>

int main(){

//variable declaration

int i,n;int x[50],max,min,sum,sum1=0;

float average,SD;

printf("Enter number of numbers to process:\n");

scanf("%d",&n);

printf("Enter the numbers to process:\n");

for(i=0;i<n;i++){scanf("%d",&x[i]);}

//Displaying numbers entered by the User

printf("Numbers entered by the User are:\n");

for(i=0;i<n;i++){printf("%4d",x[i]);

}

//computing Average valuefor(i=0;i<n;i++){

sum=sum+x[i];

average=sum/n;

//computing maximum number

max=min=x[0];for(i=0;i<n;i++){

if(x[i]>max){max=x[i];

}

}

//computing minimum number

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

if(x[i]<min){min=x[i];

}

}

//For computing Standard deviation

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

sum1=sum1+pow((x[i]-average),2);

}

SD=sqrt(sum1/n);

//Printing the Output

printf("\nStatistics Computed are:\n");

printf("Average=%f\n", average);

printf("Maximum number=%d\n",max);

printf("Minimum number=%d\n",min);

printf("Standard deviation=%f",SD);

return 0;

}

Task Seven:

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

struct Dictionary{

char Eng[40];

char Sw[40];};

int main(){

int size,I, choice, terminator;

char search[20];

printf(“Enter the size of words to use\n\n”);

printf(“Weka ukubwa wa maneno utumiayo\n”);

scanf(“%d”, &size);

struct Dictionary Array[size];

printf(“\n\nWelcome to dictionary with %d Words\n\n”, size);

for(i=0; i<size; i++) {

printf(“Enter the %d English word\n”, i+1);

scanf(“%s”, &Array[i].Eng);

printf(“Enter the %d coresponding meaning\n”, i+1);

scanf(“%s”, &Array[i].Sw);

}

printf(“\n\n”);

while(1){

printf(“\n\n”);

printf(“Enter the choice of dictionary you want\n”);

printf(“ Weka chaguzi la kamusi unayotaka kutumia\n”);

printf(“press 1. For English to Swahili\n”);

printf(“bonyeza 2. Swahili kwenda English\n”);

printf(“press 3. For displaying words\n”);

printf(“press 4. To Quit the program\n”);

scanf(“%d", &choice); printf(“\n\n”);

switch(choice) {

case 1:

printf(“Welcome to English dictionary\n\n”);

printf(“Enter the word to search\n”);

scanf(“%s", &search);

terminator = 0;

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

{

if(strcmpi(search, Array[i].Eng) == 0)//difference of characters

//if you enter HEN and you search HON

diff. ==1; the word will not be found.{

terminator++;

printf(“The meaning is: \t%s\n”, Array[i].Sw);

}

}

if(terminator == 0)

{

printf(“The word is not found in the dictionary\n”);

}

break;

case 2:

printf(“Karibu kwenye Kamusi ya Kiswahili\n\n”);

printf(“Ingiza neno la kutafuta\n”);

scanf(“%s”, &search);

terminator = 0;

for(i=0; i<size; i++) {

if(strcmpi(search, Array[i].Sw) == 0){

terminator++;

printf(“Maana yake ni: \t%s\n”, Array[i].Eng);

}

}

if(terminator == 0){

printf(“Neno unalolitafuta halipo kwenye kamusi\n”);

}

break;

case 3:

printf(“The available words in my dictionary are: %d \n”, size); printf(“the available word are:\n\n\n”);

printf(“\tEnglish ---------- Swahili\n”);

for(i=0; i<size; i++){

printf(“\t%s----------- : %s\n”);

Array[i].Eng, Array[i].Sw);

}

break;

case 4: exit(1);

break;

default : printf(“Invalid choice\n”);

}

}

}

Task Eight:

#include<stdio.h>

#include<string.h>

#include<math.h>

struct statistics{

float data[100];

char name[20];

}stat;

int main(){

int i,size,sum=0,max,min,total=0;

float avg=0,standard=0;

float data[100];

char name[20];

printf(“enter the statistics size\n “);

scanf(“%d”,&size);

printf(“enter the statistics data\n”);

for(i=1; i<=size; i++) {

scanf(“%d”, &stat.data[i]);

}

min=max=stat.data[0];

for(i=1; i<=size; i++) {

total=total+stat.data[i];

avg=total/size;

if(min>stat.data[i])

{

min=stat.data[i];

}

if(max<stat.data[i]) {

max=stat.data[i];

}

}

for(i=0;i<size;i++){

sum+=(stat.data[i]-avg)\*(stat.data[i] -avg);

standard=sqrt(sum/size);

}

printf(“enter the user name\n “);

scanf(“%d",&stat.name);

printf(“the user name is %d\n”,stat.name);

printf(“statistics data are %d\n”,stat.data[i]);

printf(“the total is %d\n”,total);

printf(“the avarage is %.2f\n”,avg);

printf(“the minimum is %d\n”,min);

printf(“the maximum is %d\n”,max);

printf(“the standard is %.2f\n”,standard);

Return 0;

}

Task nine:

// A c program to reverse a string

#include <stdio.h>

#include <string.h>

void invert\_string();

void main(){

invert\_string();}//function to receive the string inputs and reversing it.

void invert\_string(){

char arr[100];

printf(“enter the string to reverse:\n”);

gets(arr);

strrev(arr);

printf(“the reversed of the string is: \*\*\*%s\*\*\*”,arr);

return 0;

}

Task Ten:

#include<stdio.h>

#include<conio.h>

int main()

{

FILE \*fp;

char ch, fname[30], newch[500];

int i=0, j, COUNT=0;

printf("Enter the filename with extension: ");

gets(fname);

fp = fopen(fname, "r");

if(!fp)

{

printf("Error in opening the file...");

getch();

return 0;

}

printf("\nThe original file line:\n\n");

ch = getc(fp);

while(ch != EOF)

{

COUNT++;

putchar(ch);

newch[i] = ch;

i++;

ch = getc(fp);

}

printf("\n\n\n");

printf("The line in reverse order is:\n\n");

for(j=(COUNT-1); j>=0; j--)

{

ch = newch[j];

printf("%c", ch);

}

printf("\n");

getch();

return 0;

}

Task 11:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(){

//declare the variables.........................................................01

int CSS114, CSS124, CSS122, CSS118,i;

float avgCSS114 ,avgCSS124 ,avgCSS122 ,avgCSS118 ;

int score114[80],score124[80],score122[80],score118[80];

//variable initionalization..........................................................02

float sumCSS114 ,sumCSS124 ,sumCSS122,sumCSS118 ,score[350];

int y=0;

FILE \*inFile;

char student[35][80];

char best114[35],regNo[35],worst114[35];

char best124[35],regNo124[35],worst124[35];

char best122[35],regNo122[35],worst122[35];

char best118[35],regNo118[35],worst118[35];

// opening the students file containing students details.............................03

inFile = fopen("data.txt", "r");

if (inFile == NULL)

{

printf("\nFailed to open file \n");

exit(1);

}

printf("DETAILS OF STUDENTS\n");

// spaces for courses scores...............................................................04

printf("\n\n\n");

printf("regNO CSS114 CSS124 CSS122 CSS118----------------------------------------\n\n");

//Tracing the details of student from the student file......................................05

while (fscanf(inFile, "%s %d %d %d %d",&regNo, &CSS114, &CSS124, &CSS122, &CSS118) != EOF){

printf("%s\t%d\t%d\t%d\t%d\n",regNo,CSS114,CSS124,CSS122,CSS118);

//copy registration number into respective student array....................................06

strcpy(student[y],regNo);

score114[y] = CSS114;

score124[y] = CSS124;

score122[y] = CSS122;

score118[y] = CSS118;

sumCSS114 += CSS114;

sumCSS124 += CSS124;

sumCSS122 += CSS122;

sumCSS118 += CSS118;

y++;

}

int max114 = score114[0];

int min114 = score114[0];

int max124 = score124[0];

int min124 = score124[0];

int max122 = score122[0];

int min122 = score122[0];

int max118 = score118[0];

int min118 = score118[0];

// computing the maximum and the minimum score......................................07

for(i=0;i<y;i++){

if(score114[i]>max114){

max114 = score114[i];

strcpy(best114,student[i]);

}

if(score114[i]<min114){

min114 = score114[i];

strcpy(worst114,student[i]);

}

}

// testing the maximum and the minimum score using loop............................08

for(i=0;i<y;i++){

if(score122[i] > max122){

max122 = score122[i];

}

if(score118[i]>max118){

max118=score118[i];

}

if(score118[i]<min118){

min118=score118[i];

}

if(score124[i]>max124){

max124=score124[i];

}

if(score124[i]<min124){

min124=score124[i];

}

}

for(i=0;i<y;i++){

if(score122[i]<min122){

min122 = score122[i];

}

}

for(i=0;i<y;i++){

if(max114==score114[i]){

strcpy(best114,student[i]);

}

if(max122==score122[i]){

strcpy(best122,student[i]);

}

if(min122==score122[i]){

strcpy(worst122,student[i]);

}

if(max118==score118[i]){

strcpy(best118,student[i]);

}

if(min118==score118[i]){

strcpy(worst118,student[i]);

}

if(max124==score124[i]){

strcpy(best124,student[i]);

}

if(min124==score124[i]){

strcpy(worst124,student[i]);

}

}

//average coursewise..................................................................09

avgCSS114 = sumCSS114 / y;

avgCSS124 = sumCSS124 / y;

avgCSS122 = sumCSS122/y;

avgCSS118 = sumCSS118 /y;

printf("\n\n");

printf("Details coursewise..\n");

//to dispalay the results...............................................................10

printf("\nCSS114.\nAverage score : %.2f\nBest student : %s\nBest score :%d\nWorst student : %s\nWorst marks :%d\n",avgCSS114,best114,max114,worst114,min114);

printf("\nCSS124.\nAverage score : %.2f\nBest student : %s\nBest score :%d\nWorst student : %s\nWorst marks :%d\n",avgCSS124,best124,max124,worst124,min124);

printf("\nCSS122.\nAverage score : %.2f\nBest student : %s\nBest score :%d\nWorst student : %s\nWorst marks :%d\n",avgCSS122,best122,max122,worst122,min122);

printf("\nCSS118.\nAverage score : %.2f\nBest student : %s\nBest score :%d\nWorst student : %s\nWorst marks :%d\n",avgCSS118,best118,max118,worst118,min118);

fclose(inFile);

return 0;

}