**King Fahd University of Petroleum and Minerals**

**Information and Computer Science Department**

ICS 103: Computer Programming in C

**Summer Semester 2008-2009 (Term 083)**

##### Final Exam

**Time: 150 minutes Tuesday, September 1, 2009, 9:00 PM**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name:** | KEY | | | | | | | | | |
| **ID#:** |  |  |  |  |  |  |  |  |  |

PLEASE CIRCLE YOUR SECTION BELOW:

|  |  |  |  |
| --- | --- | --- | --- |
| Section | **1** | **2** | **3** |
| Time | **SUMT 8:10-9** | **SUMT 9:20-10:10** | **SUMT 9:20-10:10** |

**Note:**

* **Copying or Discussion will result in zero grade for all the students involved.**
* **Attempt all questions.**

|  |  |  |
| --- | --- | --- |
| Question # | Maximum Marks | Obtained Marks |
| 1 | 30 |  |
| 2 | 30 |  |
| 3 | 16 |  |
| 4 | 24 |  |
| Total | 100 |  |

## Good Luck

## Question 1: (30 points)

**Find the output of each of the following programs. Show your work.**

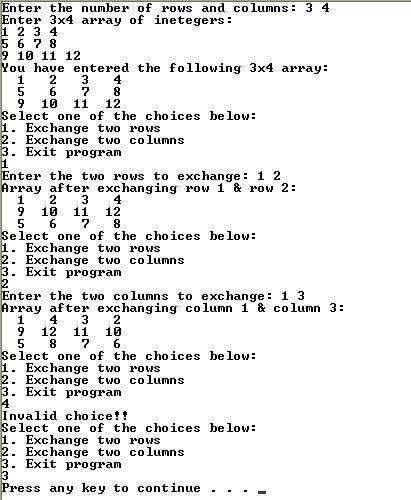
|  |  |
| --- | --- |
| Program | Output |
| 1. **6 points**   #include <stdio.h>  int test (int x[],int s,int e, int t) {  int m;    if (s > e) return -1;  m=(s + e)/2;  if (x[m] == t) return (m);  else if (x[m] < t)  return test(x,m+1,e,t);  else return test(x,s,m-1,t);  }  int main(void) {  int x[]={1,5,10,20,32,50};    printf("%d\n",test(x, 0, 5, 32));  return 0;  } | **4** |
| **6 points**  #include <stdio.h>  void test2(int a[], int n) {  int i, j = 1, k, l;  do{  k = 0;  for(i = 0; i < n - j; i++) {  if (a[i] < a[i+1]) {  l=a[i];  a[i]=a[i+1];  a[i+1]=l;  k = 1;  }  }  j++;  } while (k && j <= n-1);  }  int main(void) {  int x[]={10,5,20,15};  test2(x, 4);  for (int i=0; i<4;i++)  printf("%d ",x[i]);  printf("\n");  return 0;  } | **20 15 10 5** |
| 1. **6 points**     #include <stdio.h>  #include <string.h>  #include <ctype.h>  void test3(char a[]) {  for (int i=0; i<strlen(a);i++)  a[i]=tolower(a[i]);  }  void test4(char a[], char b[], int n) {  for (int i=0; i<n;i++)  b[i]=a[i];  }  int main(void) {  char str1[]="Ahmad is Tall", str2[]="MAD", str3[strlen(str2)];  int i,nf=1;    test3(str1);  test3(str2);    for(i=0;i<strlen(str1)&&nf;i++){  test4(&str1[i],str3,strlen(str2));  nf=strcmp(str3,str2);  }  if (nf) printf("No\n");  else printf("Yes\n");  return 0;  } | **Yes** |
| 1. **6 points**     #include <stdio.h>  #define size 3  int main(void) {  int A[size][size]={{1,2,3},{4,5,6},  {7,8,9}};    for (int i=0; i<size;i++){  for (int j=0; j<size;j++)  printf("%d ",A[j][i]);  printf("\n");    }  return 0;  } | **1 4 7**  **2 5 8**  **3 6 9** |
| 1. **6 points**     #include <stdio.h>  int main(void) {  int i;    for (i=0; i<10; i++) {  printf("\n%d ", i);  if (i < 5)  continue;  printf(" \* %d", i);  if (i > 8)  break;  }  printf("\n");    return 0;  } | **0**  **1**  **2**  **3**  **4**  **5 \* 5**  **6 \* 6**  **7 \* 7**  **8 \* 8**  **9 \* 9** |

**Question 2: (30 points)**

You are required to write a C program to read a two dimensional array of integers, exchange two rows, exchange two columns and display the array after being read and after each exchange. In your solution, you are required to do the following:

1. Ask the user to enter the number of rows and columns in the array and read them. Assume that the maximum array size is 10x10.
2. Write a function to read a two-dimensional array and use it to read the array.
3. Write a function to display a two-dimensional array and use it to display the read array.
4. Write a function to exchange two rows in the array.
5. Write a function to exchange two columns in the array.
6. Display a menu for the user asking him to select a choice between exchanging two rows, exchanging two columns or exit the program. The menu should be displayed repeatedly until the user chooses to exit the program. If an incorrect choice is entered, the message “Invalid choice !!” should be displayed.

*A sample execution of the program is shown below:*



#include <stdio.h>

#include <stdlib.h>

#define size 10

void menu(){

printf("Select one of the choices below:\n");

printf("1. Exchange two rows\n");

printf("2. Exchange two columns\n");

printf("3. Exit program \n");

}

void rarray(int a[][size],int n, int m){

printf("Enter %dx%d array of inetegers:\n",n,m);

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

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

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

}

void darray(int a[][size],int n, int m){

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

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

printf("%3d ",a[i][j]);

printf("\n");

}

}

void xchgr(int a[][size], int m, int r1, int r2){

int t;

for (int j=0;j<m;j++){

t=a[r1][j];

a[r1][j]=a[r2][j];

a[r2][j]=t;

}

}

void xchgc(int a[][size], int n, int c1, int c2){

int t;

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

t=a[i][c1];

a[i][c1]=a[i][c2];

a[i][c2]=t;

}

}

int main(void) {

int a[size][size];

int row, col, v1, v2, choice;

printf("Enter the number of rows and columns: ");

scanf("%d %d",&row, &col);

rarray(a,row,col);

printf("You have entered the following %dx%d array:\n", row, col);

darray(a,row,col);

do{

menu();

scanf("%d",&choice);

switch(choice){

case 1:

printf("Enter the two rows to exchange: ");

scanf("%d %d",&v1,&v2);

xchgr(a,col,v1,v2);

printf("Array after exchanging row %d & row %d:\n",v1, v2);

darray(a,row,col); break;

case 2:

printf("Enter the two columns to exchange: ");

scanf("%d %d",&v1,&v2);

xchgc(a,row,v1,v2);

printf("Array after exchanging column %d & column %d:\n",v1, v2);

darray(a,row,col); break;

default:

printf("Invalid choice!!\n");

}

}while (choice !=3);

system("pause");

return 0;

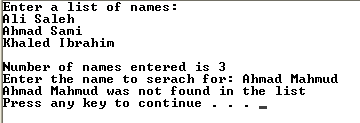
}

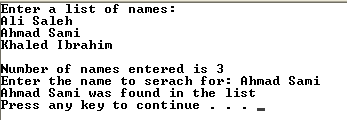
**Question 3: (16 points)**

You are required to write a C program to do the following:

1. Ask the user to enter a list of names and read them. The end of the list is indicated by the user hitting the enter key. Display the number of names entered. Assume that the maximum number of names to be entered is 40 names and the maximum name length is 80 characters.
2. Ask the user to enter a name to search for in the list. Print whether the name is found or not.

*A sample execution of the program is shown below:*





#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define RSIZE 41

#define CSIZE 81

int main() {

char Names[RSIZE][CSIZE], Name[CSIZE];

int i, ncount=0, nfound=1;

printf("Enter a list of names:\n");

gets(Name);

while (strlen(Name) !=0){

strcpy(Names[ncount++],Name);

gets(Name);

}

printf("Number of names entered is %d\n",ncount);

printf("Enter the name to serach for: ");

gets(Name);

for (i=0; i<ncount-1&&nfound!=0; i++){

nfound=strcmp(Names[i],Name);

}

if (nfound==0) printf("%s was found in the list\n", Name);

else printf("%s was not found in the list\n",Name);

system("pause");

return 0;

}

**Question 4: (24 points)**

1. Write a function that converts Cartesian coordinates to polar coordinates. That is, given *x* and *y*, the function should calculate and return *r* and** as given by

and

Hint: use atan2(y, x) to calculate .

void cartesian\_to\_polar(double x, double y, double\* r, double \*theta)

{

\*r = sqrt( x\*x+ y\*y );

\*theta = atan2(y,x);

}

1. Write a function that takes a string as an argument and returns the number of words in the string (assume words are separated by white spaces). For example, the string “C is a programming language” has 5 words.

int count\_words( char input[])

{

char delims[] = "\t\n ";

char \*token; //has to be a pointer type.

token = strtok( input, delims);

int n = 0;

while ( token != NULL ) {

n++;

token = strtok( NULL, delims );

}

return n;

}

1. Write a function that takes a square matrix as an argument and tests whether it is diagonal or not. The functions should return 1 if the input matrix is a diagonal and 0 otherwise. Hint: a diagonal matrix is a square matrix in which all elements outside the main diagonal are zeros. But the diagonal elements may or may not be zeros.

int is\_diagonal( int input[][SIZE], int s)

{

int i,j;

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

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

if( i!=j && input[i][j] !=0)

return 0;

return 1;

}