**King Fahd University of Petroleum and Minerals**

**Information and Computer Science Department**

ICS 103: Computer Programming in C

**Fall Semester 2009-2010 (Term-091)**

##### Major Exam-II

**Time:120 minutes Monday, December 28, 2009**

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

PLEASE CIRCLE YOUR SECTION BELOW:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Section | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 |
| Time | UT-9-10 | UT 8-9 | UT 10-11 | UT 11-12 | UT 1:2 | SM 9-10 | SM 7-8 | UT 7-8 |

|  |  |  |
| --- | --- | --- |
| Question # | Maximum Marks | Obtained Marks |
| 1: P1,P2,P3 | 8+7+6 |  |
| 1: P4,P5,P6 | 6+9+6 |  |
| 2 | 16 |  |
| 3 | 10 |  |
| 4 | 15 |  |
| 5 | 17 |  |
| Total | 100 |  |

 **Notes.** 1. Make sure you have **EIGHT** pages including the cover page.

 2. Closed book and notes

 3. Write clearly, briefly and precisely

 4. Cheating will result in ZERO grade

##  Good Luck

## Question 1: (42 points)

|  |  |
| --- | --- |
| **P1: 8 points**Given an array list x[size]={2, 8, 1, 4, 3, 5} with size 6. Write a C program to display smallest and largest number from an array list. | #define size 6int main(){  int i,max, min;  int x[size]={2, 8, 1, 4, 3, 5}; max=x[0]; min=x[0]; for(i=0;i<size;i++) {  if(max <= x[i])  max = x[i]; else if (min > x[i]) min = x[i];  }  printf("Max number: %d\n",max);  printf("Min number: %d\n",min);} |
| #include <stdio.h> // **P2: 7 points**#include<stdlib.h>int main(void){ int i, j, x=1; for(i=1; i<5; ++i) for(j=1; j<i; ++j) { x += (i+j-1); printf("%d\n", x); } printf("\nx=%d", x ); system("pause");  return 0; } | **3****6****10****14****19****25****x=25** |

|  |  |
| --- | --- |
| #include <stdio.h> // **P3: 6 points** void fun1(int n);void fun2(int n);int main(void){  fun2(3);  return 0; } void fun1(int n){ for (int i=n; i>0; i--) printf("%d",i); printf("\n");}void fun2(int n){ if (n==0) return; else { fun1(n); fun2(n-1); fun1(n); } } | **321****21****1****1****21****321** |

|  |  |
| --- | --- |
| #include <stdio.h> // **P4 6 points** int main (){ int x = 5, y = 15, z; int \*p1, \*p2; p1 = &x;  p2 = &y;   \*p1 = 10;  \*p2 = \*p1; z = x + y; p1 = p2;  \*p1 = 30;   printf("x = %d, y = %d, z = %d \n", x, y,z); return 0;} | **x = 10, y = 30, z = 20** |

|  |  |
| --- | --- |
| #include <stdio.h> // **P5: 9 points**#define SIZE 6void last(int \*x, int y, int n);int main() { int a[SIZE],i; a[0]=2;a[2]=5;a[4]=8; for(i=1;i< SIZE;i=i+2) last(&a[i], a[i-1], i); for(i=4;i>=0;i=i-2) last(&a[i], a[i+1], i); for(i=0;i<SIZE;i++) printf("%d\n",a[i]); return 0;}void last (int \*x, int y, int n) {  if( n%2 == 0) \*x = y+n; else \*x = y - n;} | **1****1****4****2****7****3** |

|  |  |
| --- | --- |
| #include<stdio.h> // **P6 6 points**void check(int \*a,int \*b, int i){ if(i%3==0) \*a=\*b+1; else if (i%2 == 0) \*a=\*b-1; else \*a = \*b; }int main(void){ int i,x[4]={2,3,7,8}; int y[4]; for (i = 0 ;i<4;i++ ) check(&y[i],&x[3 - i],i);  for (i = 0 ;i<4;i++ ) printf("%d\n",y[i]); return 0;} | **9****7****2****3** |

**Question 2: (16 points )**

Write a C program to open an input file named ***myinput.txt*** and count the number of vowel characters (i.e. ‘a’, ‘e’, ‘i', ‘o’, ‘u’, 'A', 'E', 'I', 'O' and 'U') both lower and upper cases, the number of digit characters (i.e. ‘0’ to ‘9’), the number of consonants (i.e. non-vowel letters like ‘b’, 'c', 'd', 'f'...’z’ and ‘B’, 'C', 'D', 'F',…,’Z’). The results must be printed in an output file named ***myoutput.txt***. Your program should handle file not found error. Minimize the number of statements used in your program.

Note: Assume that the input file ***myinput.txt contains only digits and letters (no other characters)****.*

**Below are samples of input and output files.**

***myinput.txt*** file ***myoutput.txt*** file

  

**#include <stdio.h>**

**#include <stdio.h>**

**int main(void){**

**FILE \*infile, \*outfile;**

**char c;**

**int consonants=0, digits=0, vowels=0, status;**

**infile = fopen("myinput.txt","r");**

**outfile = fopen("myoutput.txt","w");**

**if (infile == NULL){**

 **printf("Cannot open myinput.txt for reading \n");**

 **exit(1);**

**}**

**status = fscanf(infile,"%c",&c);**

**while ( status != EOF ) {**

**if (c>='0' && c<='9')**

 **digits++;**

**else if ( c == 'a' || c=='A' || c=='e' || c=='E' || c == 'i' ||c ==**

**'I'|| c == 'o' || c == 'O' || c == 'U' || c == 'u')**

 **vowels++;**

**else if(c >= 'a' && c <= 'z' || c >= 'A' && c <= 'Z')**

 **consonants++;**

**status = fscanf(infile,"%c",&c);**

**}**

**fprintf(outfile,"Number of vowel characters = %d \n",vowels);**

**fprintf(outfile,"Number of numeric characters = %d \n",digits);**

**fprintf(outfile,"Number of consonants = %d \n",consonants);**

**fclose(infile);**

**fclose(outfile);**

**return 0;**

**}**

**Question 3: (10 pts )**

Write a function **countodd** thatreceives 2 input arguments n1 and n2 of type int. n1 and n2 represent the end values of the interval [n1,n2] (n1 and n2 are included in the interval). The function will return how many odd values are inside the interval. It does not print anything.

**Note: Write the function definition only. No main function.**

int countodd (int n1 , int n2) {

int i,count=0;

for( i=n1 ; i<= n2; i++) {

 if(i%2 != 0)

 count++;

}

return count;

}

**Question 4: (15 points )**

Write a function, ***dist\_slope***, that takes the coordinates of two points [(x1, y1) and (x2, y2)] as input arguments and returns the distance between the two points and the slope of the line connecting the two points using **output arguments (pointers)**.

 

Complete the following main function so that the function ***dist\_slope***  is called and the results are printed as shown below.

**Note: You are not allowed to use printf inside the function *dist\_slope***



**#include <stdio.h>**

**#include <math.h>**

**void dist\_slope(double x1, double y1, double x2, double y2, double \*dist, double \*slope);**

**int main() {**

**double x1=2.34,y1=7.89,x2=3.11,y2=16.14,dist,slope;**

**dist\_slope(x1,y1,x2,y2,&dist,&slope);**

**printf("distance=%.2f\n",dist);**

**printf("slope=%.2f\n",slope);**

**return 0;**

**}**

**//function definition after this line**

**void dist\_slope(double x1, double y1, double x2, double y2, double \*dist, double \*slope){**

**\*slope=(y2-y1)/(x2-x1);**

**\*dist=sqrt(pow(x2-x1,2)+pow(y2-y1,2));**

**}**

**Question 5: (17 points)**

Write a C program that declares one integer array for storing student ids called ***id\_array*** and a double array for storing student scores called ***scores\_array***. Your program starts by asking the user to enter the number of students and then reads the student ids and their scores and stores them in the corresponding arrays. The program then computes and displays the average, the highest mark along with the student id that scored the highest, and the lowest mark along with the student id that scored the lowest. The program then displays the ids of students who are above average. The results should be displayed correct to two decimal places. Assume that the maximum expected number of students is 100. The maximum number of students should be defined as a constant.

A sample run of your program must look like this:

Enter the number of students: 5

Enter the student ids and their scores:

200963670 93

200992650 97

200914730 90.5

200945730 76.5

200894600 80

Average = 87.40

Highest is 200992650 with a score of 97.00

Lowest is 200945730 with a score of 76.50

Students above average are:

200963670

200992650

200914730

**#include <stdio.h>**

**#include <stdlib.h>**

**#define SIZE 100 // 0.5 mark**

**int main(void){**

 **int id\_array[SIZE]; // 0.5 mark**

 **int n, i, max=0, min=0; // 0.5 mark**

 **double scores\_array[SIZE]; // 0.5 mark**

 **double sum=0, avg; // 0.5 mark**

 **printf("Enter the number of students: "); // 0.5 mark**

 **scanf("%d", &n); // 0.5 mark**

 **printf("Enter the student ids and their scores:\n"); //0.5 mark**

 **for (i=0; i<n; i++){ // 1 mark**

 **scanf("%d%lf", &id\_array[i], &scores\_array[i]); // 1.5 marks**

 **sum += scores\_array[i]; // 1 mark**

 **if (scores\_array[i] > scores\_array[max]) max = i; //1.5 marks**

 **else if (scores\_array[i] < scores\_array[min]) min = i;//1.5 m**

 **}**

 **avg = sum / n; // 0.5 mark**

 **printf("Average = %.2f\n", sum/n); // 0.5 mark**

 **printf("Highest is %d with a score of %.2f\n", id\_array[max], scores\_array[max]); // 1.5 marks**

 **printf("Lowest is %d with a score of %.2f\n", id\_array[min], scores\_array[min]); // 1.5 marks**

 **printf("Students above average are:\n"); // 0.5 mark**

 **for (i=0; i<n; i++) // 1 mark**

 **if (scores\_array[i] > avg) // 0.5 mark**

 **printf("%d\n",id\_array[i]); // 0.5 mark**

 **system("pause");**

 **return 0;**

**}**