Information and Computer Science Department

Fall Semester 141

ICS 103 – Computer Programming in C

Final Exam

Tuesday, January 06, 2015

Duration: 140 minutes

|  |  |
| --- | --- |
| Name |  |
| ID |  |  |  |  |  |  |  |  |  |
| Section |  |
| Instructor |  |

|  |  |  |
| --- | --- | --- |
| Question # | MaximumGrade | ObtainedGrade |
| 1 | 16 |  |
| 2 | 24 |  |
| 3 | 15 |  |
| 4 | 10 |  |
| 5 | 15 |  |
| 6 | 20 |  |
| Total | 100 |  |

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Question 1 [16 points]

Select the right answers for the following multiple choice questions. Write your answers in the given table at the end of this question on the page 5.

1. Consider the following code fragment:

char line[90];

int i;

gets(line);

What is the correct loop to write after gets(line)that allows us to display just the characters typed by the user ?

1. for(i=0;i<90;i++) printf("%c",line[i]);
2. for(i=0;i<89;i++) printf("%c",line[i]);
3. for(i=0;line[i]!= '\0';i++) printf("%c",line[i]);
4. for(i=0;i!='\0';i++) printf("%c",line[i]);
5. What is correct prototype for the function “fun” that takes an input argument, and does not return any value?
6. int fun (void);
7. int fun;
8. fun (int);
9. void fun (int);
10. Which of the following is a valid function call (assuming the function exists)?
11. func( );
12. int func( );
13. func x, y;
14. func;
15. In C language, call by reference is also known as:
16. Call by address or Call by value
17. Call by address
18. Call by value
19. None of the above
20. The following statement represents assignment of a value to the element of a **full** 2-D array located at the bottom right corner.

 mat[ 2 ][ 4 ]= 10;

then array **mat** must be declared with

1. 5 columns and 3 rows
2. 3 columns and 5 rows
3. 4 columns and 6 rows
4. 6 columns and 4 rows
5. To assign values to all the elements of an array vect of size 5 in **reverse** order, the correct statements are:
6. for (i = 0; i < 5; ++i) scanf("%d",&vect[5]);
7. for (i = 5; i >= 0; ++i) scanf("%d",&vect[i]);
8. for (i = 4; i >= 0; --i) scanf("%d",&vect[i]);
9. for (i = 0; i <= 5; ++i) scanf("%d",&vect[5-i]);
10. In C, if an array is passed as an argument to a function, what actually gets passed?
	1. The values of all elements in the array
	2. The value of the first element of the array
	3. The Address of the last element of array
	4. The Address of first element of the array
11. Consider the following array:

 {8, 7, 5, 4, 5, 6, 4, 3}.

When calling linearSearch function (studied in class) using a target value (value to search for) of 5, the value returned by the function is

* 1. 3
	2. 5
	3. 2
	4. 4
1. Consider the following array:

 {8, 7, 5, 4, 5, 6, 4, 3}.

When calling linearSearch function (studied in class) using a target value (value to search for) of 10, the value returned by the function is

* 1. 0
	2. -1
	3. 8
	4. false
1. Consider the *selection sort* function. Select the correct order of the elements of the array after each pass to sort the following array: {15,20,10,18}
	1. {10, 20,15,18} – {10,15,20,18} – {10,15,18,20}
	2. {15,10,18,20} – {10,15,18,20} – {10,15,18,20}
	3. {10,18,15,20} – {10,15,18,20} – {10,15,18,20}
	4. {10,15,20,18} – {10,15,20,18} – {10,15,18,20}
2. What is the output of the following code fragment?

char final[2][10]={"ics103","141"};

final[1][3]='\0';

printf("%d %d",strlen(final[0]),strlen(final[1]));

1. 9 9
2. 6 4
3. 3 3
4. 6 3
5. Consider the following declaration:

 char s1[30];

 The correct way to assign string "ICS" to s1 is

1. s1 = "ICS";
2. strcpy("ICS",s1);
3. strcpy(s1, "ICS");
4. s1[30]= "ICS";
5. Consider the following declaration:

 char s1[30] = "hi", s2[20] = "world", s3[10]= "";

 The correct way to have string "hiworld" in s3 is

* 1. s3 = s1 + s2;
	2. strcpy(s3,s1); strcat(s3,s2);
	3. strcat(s3,s1); strcpy(s3,s2);
	4. strcpy(s3,s1); strcpy(s3,s2);
1. Which of the following can be used to declare and initialize a 2-D array without any error or warning?
	1. int A[][4] = {{1,2,3},{3,4}};
	2. int A[2][] = {{1,2,3},{4,5,6}};
	3. int A[][] = {{1,2},{3,4}};
	4. int A[3][3]={{1,2},{3,4,5,6}};
2. The correct prototype for a function receiving array A with values and copying them in array B in reverse order is:
3. void CopyArray(int A[], int \*B[], int n);
4. void CopyArray (int \*A[], int \*B[], int n);
5. void CopyArray (int \*A[], int B[], int n);
6. void CopyArray (int A[], int B[], int n);
7. What is the output of the following C code segment?

if(strcmp("10","2")>0)

 printf("bigger");

else

 printf("smaller");

1. bigger
2. smaller
3. error the first argument of strcmp must be an array of type char
4. error both arguments of strcmp must be arrays of type char

Answers AAAAAAAAAAAAAAAAAAAAAAAAAAAA

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1: C | 2: D | 3: A | 4: B | 5: A | 6: C | 7: D | 8: C |
| 9: B | 10: A | 11: D | 12: C | 13: B | 14: A | 15: D | 16: B |

Question 2 [24 points]

What is the output of the following C programs?

|  |  |
| --- | --- |
| #include <stdio.h> //6 points int main() { int i, a[5] = {2, 1, 3, 2, 1}; for(i=1; i <= 5; i++)  if(i<4) a[i] = a[i-1] + a[i+1]; else a[i%5]=a[i-1]+a[(i+1)%5];for(i=0;i<5;i++) printf("%d ",a[i]);return 0; } | 15 5 7 8 10 |

|  |  |
| --- | --- |
| #include <stdio.h> //6 pointsint main() {int i,j, k ;int b[3][3]={{5,3,9},{4,1,2},{6,7,8}};for (k=0;k<3;k++) { j=0; for (i=k;i>=0;i--){ printf("%d ",b[i][j]); j++; } printf("\n"); }return 0;} | 54 36 1 9 |
| #include <stdio.h> // 6 pointsint main() {int a=2,b=3,c;int \*p1,\*p2;p1=&a;p2=&b;\*p1=2+\*p2;\*p2=a+3;c=a+b;printf("%d %d %d",a,b,c);return 0;} | 5 8 13 |
| #include <stdio.h> // 6 points int main(){ int count=0,m,n; m=2; do{ n=9; while(n>=4){ count++; n=n-2; } m=m+3; }while(m<=5); printf("%d %d %d\n",m,n,count);return 0;} | 8 3 6 |

Question 3 [15 points]

Write a C function that receives 1-D array of real values and **n** representing the number of values in the array. The function returns (**not prints**) 2 results using output arguments .

These 2 results represent the absolute maximum difference between **two adjacent** elements and the location (index of the smaller of the 2 adjacent elements).

For example if the received array is {3.1, 5.6, 8.2, 9.3, 5.2}, the differences are {2.5,2.6,1.1,-4.1}. The function will return 4.1 (maximum absolute difference) and 3 (index of 9.3).

Note: write function definition only

void maxdif(double x[], int n, double \*absmax, int \*index) {

 int i;

 double dif;

 \*absmax=fabs(x[1]-x[0]);

 \*index=0;

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

 dif=fabs(x[i+1]-x[i]);

 if(dif>\*absmax){

 \*absmax=dif;

 \*index=i;

 }

 }

}

Question 4 [10 points]

Write a function that receives 2 1-D arrays of real values and **n** representing number of elements in each array (vector). The function computes and returns (**not print**) their dot product.

Background: The dot product of two vectors a = [a0, a1, a2, … , an-1] and

b = [b0, b1, b2, … , bn-1] is defined as:

$$dot-product=\sum\_{i=0}^{n-1}a\_{i}×b\_{i}$$

For example if a=[3.4,-5.2,6] and b=[2.5,1.6,-2.9]

Dot product=3.4x2.5+(-5.2)x1.6+6x(-2.9)

double dot\_product (double a[], double b[], int n) {

double sum;

int i;

sum=0;

fori=0;i<n;i++)

 sum=sum+a[i]\*b[i];

return sum;

}

Question 5 [15 points]

Write a C function that receives 2 strings st1 and st2, the it will modify st1 by inserting the characters of string st2 at the beginning of st1 but in reverse order.

For example if st1 contains "ICS103" and st2 contains "FINAL", st1 becomes "LANIFICS103".

The function prototype is

void insertreverse (char st1[], char st2[]);

NOTE: the only predefined function you are allowed to use is strlen. You have to move the characters one by one with a loop.

Hint: you need to create space at the beginning of string st1, then copy characters of st2 in reverse order.

void insertreverse(char st1[], char st2[]){

 int len1=strlen(st1); // find how many characters are in each strring

 int len2=strlen(st2),i;

 for(i=len1-1;i>=0;i--) // move characters of st1 len2 spaces to the right to create place for st2

 st1[i+len2]=st1[i];

 st1[len1+len2]='\0'; add Null character

 for(i=0;i<len2;i++) copy st2 characters at beginning of st1 in reverse order

 st1[i]=st2[len2-1-i];

}

Strcat and strcpy are not allowed

Question 6 [20 points]

A certain Land data is given as a 2-D array (matrix) of integer values. Each element of the array represents the altitude (height) at that position. This data is stored in a file “input.txt” as shown below. The first 2 values in the input file represent the actual number of rows and columns of the matrix, followed by the values of the matrix stored row-wise.

Write a complete C program that reads the data from “input.txt” file into a 2-D array (matrix) with ROWS rows and COLS columns to be defined as constants with a value of 100. Then it will find and print on the screen the peaks with their locations (row and column index). You need to handle “file not found” case.

A peak is each interior element of the array whose value is greater than all the values of its 8 surrounding elements.

Note that the elements of the first and last rows and columns are not considered in the search because they don’t have 8 surrounding elements.

For example, if your program uses the file shown below as an input, it will create the 4 by 6 array shown on the right side. The search will be in the highlighted part of the array. The output generated by the program for this example is also shown.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 2 | 1 | 0 |
| 6 | 1 | 6 | 1 | 13 | 5 |
| 3 | 10 | 6 | 5 | 9 | 12 |
| 6 | 2 | 3 | 12 | 5 | 1 |



Question 6 [20 points]

#include <stdio.h>

#include <stdlib.h>

#define ROWS 100

#define COLS 100

int main() {

 int mat[ROWS][COLS];

 int i,j,m,n,rows,cols,count;

FILE \*in;

in=fopen("input.txt","r");

if(in==NULL) {

 printf(“File not found\n”);

 exit(1);

}

fscanf(in,"%d%d",&rows,&cols);

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

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

 fscanf(in,"%d",&mat[i][j]);

for(i=1;i<rows-1;i++){ // loop inside the inner region

 for(j=1;j<cols-1;j++){

 count=0;

 for(m=i-1;m<=i+1;m++){ // loop around the 8 neighbors of mat[i][j]

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

 if(mat[i][j]>mat[m][n])

 count++;

 }

 if(count==8) check if mat[i][j] is bigger than all the 8 neighbors

 printf("element %d,%d is a peak its value is %d\n",i,j,mat[i][j]);

 }

 }

fclose(in);

return 0;

}

Question 6 [20 points]

Other version using IF with 8 comparisons

#include <stdio.h>

#define ROWS 100

#define COLS 100

int main() {

 int mat[ROWS][COLS];

 int i,j,m,n,rows,cols;

FILE \*in;

in=fopen("input.txt","r");

if(in==NULL) {

 printf(“File not found\n”);

 exit(1);

}

fscanf(in,"%d%d",&rows,&cols);

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

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

 fscanf(in,"%d",&mat[i][j]);

 for(i=1;i<rows-1;i++){

 for(j=1;j<cols-1;j++){

// compare mat[i][j] with its 8 neighbors

 if( mat[i][j]>mat[i-1][j-1] && mat[i][j] > mat[i-1][j] && mat[i][j] > mat[i-1][j+1]

 && mat[i][j]>mat[i][j-1] && mat[i][j]>mat[i][j+1]

 && mat[i][j] > mat[i+1][j-1] && mat[i][j] > mat[i+1][j] &&

 mat[i][j] > mat[i+1][j+1])

 printf("element %d,%d is a peak its value is %d\n",i,j,mat[i][j]);

 }

 }

fclose(in);

return 0;

 }