Information and Computer Science Department

Second Semester 153

ICS 103 – Computer Programming in C

Final Examination

Wednesday, August 31, 2016

Duration: 120 minutes

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | KEY | Serial # |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID#: |  |  |  |  |  |  |  |  |  | Section# |  |

**Instructions:**

1. Answer all questions. Make sure your answers are clear and readable.
2. Make sure there are **5** questions in **11** pages.
3. The exam is closed book and closed notes. No calculators or any helping aides are allowed. Make sure

to turn off your mobile phone and keep it in your pocket.

1. If there is no space on the front of a question’s page, use the back of the page. Indicate this clearly.

|  |  |  |  |
| --- | --- | --- | --- |
| Question # | Maximum Grade | Obtained Grade | Remarks |
| 1 | 12 |  |  |
| 2 | 32 |  |  |
| 3 | 18 |  |  |
| 4 | 18 |  |  |
| 5 | 20 |  |  |
| Total | 100 |  |  |

**Question # 1 [12 points]**

1. [4 points] Convert the following C if-statement into equivalent switch-statements. Assume that the variables **k** and **m** are integer variables and that the variable **x** is initialized.

|  |  |
| --- | --- |
| if(k == 8)  x += 3;  else if (k == 2|| k == 4)  x -= 3;  else {  if(m == 4)  x /= 5;  else if(m == 8)  x \*= 2;  } | switch(k) {  case 8: x += 3; break;  case 2:  case 4: x -= 3; break;  default: switch(m) {  case 4: x /= 5; break;  case 8: x \*= 2; break;  }  } |

1. [4 points] Convert the following program fragment into an equivalent fragment that uses **nested for-loops** instead of **nested while-loops**:

|  |  |
| --- | --- |
| double sum = 0;  int i = 1, j;  while(i < 15){  j = i;  while(j <= 20){  sum = sum + i + j;  j++;  }  printf("sum = %0.1f\n",sum);  i = i + 2;  } | double sum = 0 ;  int i, j;  for(i = 1; i < 15; i = i + 2) {  for(j = i; j <= 20; j++)  sum = sum + i + j;  printf("sum = %0.1f\n",sum);  } |

1. [4 points] Trace the Selection sort algorithm in sorting the array **25, 4, 60, 1, 8, 2, 15** in increasing order (i.e., from lowest to highest) by showing the contents of the array at the end of each of the first two passes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Pass1 |  | Pass2 |
| 25 |  | **1** |  | 1 |
| 4 |  | 4 |  | **2** |
| 60 |  | 60 |  | 60 |
| 1 |  | 25 |  | 25 |
| 8 |  | 8 |  | 8 |
| 2 |  | 2 |  | 4 |
| 15 |  | 15 |  | 15 |

**Question # 2 [32 points]**

Write the output of each of the following C programs or program fragments:

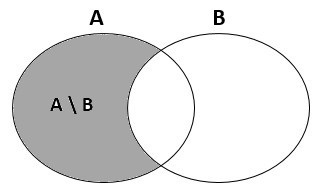
|  |  |
| --- | --- |
| [3 points]  What is the output if the input is:  RIYADH IS A BIG CITY  char str1[81], str2[81];  scanf("%s", str2);  gets(str1);  puts(str1);  printf("%s", str2);  printf("%s", str2); | **Output:**  IS A BIG CITY  RIYADHRIYADH |
| [5 points]  #include <stdio.h>  int main(void){  int matrix[][4] = {{5, 20, 8, 15}, {9, 6, 40, 2}, {3, 70, 1, 10}}, k, m;  for(m = 3; m >= 0; m--){  for(k = 0; k <= 2; k++){  printf("%2d ", matrix[k][m]);  }  printf("\n");  }  return 0;  } | **Output:**  15 2 10  8 40 1  20 6 70  5 9 3 |
| [3 points]  int a = 5, b = 2, \*p1, \*p2;  p1 = &a;  p2 = &b;  \*p1 = \*p2 + 8;  b = \*p1 \* \*p2;  printf("%d %d", a, \*p2); | **Output:**  10 20 |
| [5 points]  #include <stdio.h>  int funcn(int a, int b, int\* c, int\* d);  int main(void) {  int x = 7,y = 3, z[] = {6, 3};  printf("%d\n",funcn(x, z[0], &z[1], &y));  printf("%d %d %d %d",x, z[0], z[1], y);  return 0;  }  int funcn(int a, int b, int\* c, int\* d) {  a = 3 \* b;  b = 4;  \*c = 2;  \*d = 6 + \*c ;  return a + b + \*d;  } | **Output:**  30  7 6 2 8 |
| [7 points]  #include <stdio.h>  #include <string.h>  int main(void){  char strings[][80] = {"THAT", "VERY", "COLD", "MONTH"};  char str[80] = "";  int k, m = 0, index;  for(k = 0; k <= 3; k++){  index = strlen(strings[k]) - 2;  str[m] = strings[k][index];  printf("%s\n", str);  m++;  }    for(m = strlen(str) - 1; m >= 0; m--){  printf("%c", str[m]);  }  return 0;  } | **Output:**  A  AR  ARL  ARLT  TLRA |
| [5 points]  #include <stdio.h>  #include <ctype.h>  #include <string.h>  int main(void){  //Note: There is one space between 9 and 4 in string str  char str[ ] = "9 47\nBc\tRod";  int k;  for(k = strlen(str) - 1; k >= 0; k--){  if(isupper(str[k])){  str[k] = tolower(str[k]);  printf("%c\n", str[k]);  }  else if(isspace(str[k]))  str[k] = '\*';  else if(str[k] == '9')  str[k] = '0';  else if(isdigit(str[k]))  str[k] = str[k] + 1;  }  puts(str);  return 0;  } | **Output:**  **r**  **b**  **0\*58\*bc\*rod** |

|  |  |
| --- | --- |
| [4 points]  #include <stdio.h>  int function(int n){  if(n == 1 || n == 0)  return n;  else{  printf("%d ", n);  return n + function(n - 2);  }  }  int main(void){  printf("%d", function(10));  return 0;  } | **Output:**  **10 8 6 4 2 30** |

**Question # 3 [18 points: 16 + 2]**

1. The set difference of sets **A** and **B**, denoted by **A \ B**, is the set of all elements of **A** that do not belong to set **B**:

A \ B = { x such that x A and x



A one-dimensional array with no repeated elements can represent a set. Write a void function **setDifference** that has six parameters: three one-dimensional integer array parameters **A**, **B, and C**, the size of array **A**, the size of array **B**, and a parameter **numElements**. The function receives **A** and **B** as input parameters and it returns the set difference **A / B** in the array **C** and the number of elements of **C** in the parameter **numElements**.

Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| Array A | Array B | Array C | numElements |
| {10, 20, 30 } | {10, 30, 45, 20} | (empty set) | 0 |
| {1, 2, 3 , 4, 5} | {3, 2, 4, 6} | {1, 5} | 2 |
| {25, 70, 12, 8, 3, 4} | {3, 70, 16, 25 } | {12, 8, 4} | 3 |

**Note:**

* Your function must be general.
* Assume that both arrays **A** and **B** are filled with values and that each has no repeated element.
* Your function must not contain **scanf** and **printf** calls.
* DON’T WRITE THE **main** FUNCTION.

**void setDifference(int A[], int B[], int C[], int sizeA, int sizeB, int\* numElements){**

**int k, m, count = 0, isPresentInB;**

**for(k = 0; k <= sizeA - 1; k++){**

**isPresentInB = 0;**

**for(m = 0; m <= sizeB - 1; m++){**

**if(A[k] == B[m]){**

**isPresentInB = 1;**

**break;**

**}**

**}**

**if(!isPresentInB){**

**C[count] = A[k];**

**count++;**

**}**

**}**

**\*numElements = count;**

**}**

1. Assuming that integer arrays **array1** and **array2** are initialized and the number of elements in these arrays, namely, **numArray1**, **numArray2** are also initialized, also an integer array **array3** and an integer **numArray3** are declared, write an appropriate call to the function **setDifference** that you were required to write in 3(a).

**Note:** You are only required to write the function call; leave the rest of the page blank.

**setDifference(array1, array2, array3, numArray1, numArray2, &numArray3);**

**Question # 4 [18 points]**

Write an integer function **getSubarray** that has the following **five** parameters:

* Two 2D-arrays **x** and **y** of type **double** with the same dimensions.
* The number of filled rows and the number of filled columns in the array **x**.
* An integer column number **N**.

The function **getSubarray** returns in the array **y** the subarray of **x** thatstarts from column N to the last column of array **x**. Your function must return **0** without modifying the empty array **y** if **N** is not a valid column number of array **x**; otherwise it modifies array **y** and then returns **1**.

**Example:**

If array **x** is:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1.0 | 5.0 | 9.0 | 13.0 | 11.0 | 30.0 |  |
| 2.0 | 6.0 | 10.0 | 23.0 | 22.0 | 80.0 |  |
| 3.0 | 7.0 | 11.0 | 35.0 | 33.0 | 12.0 |  |
| 4.0 | 8.0 | 12.0 | 43.0 | 44.0 | 40.0 |  |
|  |  |  |  |  |  |  |

and N is 3, then the modified array **y** is:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 13.0 | 11.0 | 30.0 |  |  |  |  |
| 23.0 | 22.0 | 80.0 |  |  |  |  |
| 35.0 | 33.0 | 12.0 |  |  |  |  |
| 43.0 | 44.0 | 40.0 |  |  |  |  |

**Note:**

* Assume a constant **MAXCOLS** that stores the maximum number of columns in array **x** and **y** is defined outside all functions.
* The array **y** is passed as an empty un-initialized array.
* Your function must be general, it must not contain **scanf** and **printf** calls, and it must use appropriate **nested-loop**.
* DON’T WRITE THE **main** FUNCTION.

**int getSubarray(double x[][MAXCOLS], double y[][MAXCOLS], int nrows, int ncols, int N){**

**if(N < 0 || N > ncols)**

**return 0;**

**int k, r, c;**

**c = 0;**

**for(k = N; k <= ncols - 1; k++){**

**for(r = 0; r <= nrows - 1; r++){**

**y[r][c] = x[r][k];**

**}**

**c++;**

**}**

**return 1;**

**}**

**Question # 5 [20 points]**

Each student in a class has done at least one quiz. The quiz numeric grades are stored in a file **input.txt** that has the form:

|  |
| --- |
| 201500001 34 45 60 70  201500002 15 30  201500003 56  201500004 20 40 50  201500005 37 |

Where each line contains a student ID followed by the numeric grades of his quizzes.

Write a complete C program that reads the **input.txt** file and outputs on each line of an output file **output.txt**, the ID of a student, followed by the number of quizzes he has done and his quiz average:

|  |
| --- |
| ID# QUIZ COUNT AVERAGE  201500001 4 52.3  201500002 2 22.5  201500003 1 56.0  201500004 3 36.7  201500005 1 37.0 |

**Hint**: You may use the following standard functions in your program:

|  |  |  |
| --- | --- | --- |
| **function** | **What the function does** | **Example** |
| int atoi(char\* string) | Converts a string to an integer value | atoi(**"**457**"**) returns 457 |
| double atof(char\* string) | Converts a string to a double value | atof(**"**23.42**"**) returns 23.42 |

The prototypes of both **atoi** and **atof** are defined in the header file **stdlib.h**

Note:

* Your program must be general, it must work for an input file of any number of lines.
* Assume that the maximum line size in the text-file is 80.

**#include <stdio.h>**

**#include <stdlib.h> // atoi, atof, exit**

**#include <string.h>**

**int main(void){**

**FILE \*infilePtr, \*outfilePtr;**

**int id, gradeCount = 0;**

**double grade, gradeSum, average;**

**char line[81], delimeters[] = " \t\n";**

**char \*token;**

**if((infilePtr = fopen("input.txt", "r"))== NULL){**

**printf("Error: Cannot open input.tx");**

**exit(1);**

**}**

**outfilePtr = fopen("output2.txt", "w");**

**fprintf(outfilePtr, "%9s%15s%11s\n", "ID#", "QUIZ COUNT", "AVERAGE");**

**while( fgets(line, 81, infilePtr) != NULL){**

**gradeSum = 0;**

**gradeCount = 0;**

**token = strtok(line, delimeters);**

**id = atoi(token);**

**token = strtok(NULL, delimeters);**

**do{**

**gradeSum += atof(token);**

**gradeCount++;**

**token = strtok(NULL, delimeters);**

**} while(token != NULL);**

**average = gradeSum / gradeCount;**

**fprintf(outfilePtr, "%9d%15d%11.1f\n", id, gradeCount, average);**

**}**

**fclose(infilePtr);**

**fclose(outfilePtr);**

**printf("Output file output2.txt created . . .");**

**return 0;**

**}**