|  |
| --- |
| King Fahd university of petroleum & Minerals |
| ICS 103 Final Exam |
| For semester 102 |
| Sunday, June 12, 2011

|  |  |
| --- | --- |
| **Name** |  |
| **ID** |  |
| **Section Instructor** |  |
| **Section Number or class time** |  |

 |
|  |
|

|  |  |  |
| --- | --- | --- |
| **Question 1** | **24** |  |
| **Question 2** | **36** |  |
| **Question 3** | **40** |  |
| **Total** | **100** |  |

 |

*Version B*

# Question 1: Find the output of each code block of the following

**24pts = 4 + 4 + 3 + 4 + 5 + 4**

|  |  |
| --- | --- |
| char str[] = “This is interesting”;char delims[] = “i”;char \*token; token = strtok( str, delims );while ( token != NULL ) { puts(token); token = strtok( NULL, delims );} |  |
| void fun(int \*a, int \*b, int c, int d){ int x; x=\*a; \*a = c; d = \*b; c = x; }int main(void){ int p=10,q=15,r=20, s=25; fun(&p, &q, r, s); printf("%d %d %d %d", p, q, r, s); printf("\n");} |  |
| int A[8] = {2,1,3,6,5};int j = 7,i;for(i=0 ; i<=7 ; i++) printf("%d ", A[j] - A[i]);  |  |
| int x[] = {4,6};int y[] = {10,4};int i,j;for(i=0 ; i<2 ; i++){  for(j=0 ; j<2 ; j++) printf(“%d ”, x[i] \* y[j]); printf(“\n”);} |  |
| int A[8], n=8;for(i=0; i < n; i++) if (i%2==0) A[i] = 5\*i; else A[n-i] = -5\*i;for(i=0; i < n; i++) printf(“%d\n”,A[i]); |  |
| char names[][80] = {“Good Morning”, “Hello”, “Hi”};int i;for(i=0; i<2; i++)printf(“%d %c\n”,strlen(names[i]), names[i][2] ); |  |

# Question 2 (36 pts = 3 pts each): Choose the correct answer of the following multiple choice questions

***Your answers***

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***10*** | ***11*** | ***12*** |
|  |  |  |  |  |  |  |  |  |  |  |  |

**1. Consider the following incomplete code**

char name[50] = "my name is ahmad"; int i;

 for(i=0 ; i< strlen(name) ; i++)

 {

 if( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )

 {

 name[i] = toupper(name[i]);

 }

 }

**In order for the previous code to convert every first letter of each word of a string to upper case *without any errors*, the if condition needs to be :**

1. isspace(name[i-1]) && i+1 < strlen(name)
2. i>0 && isspace(name[i-1]))
3. isspace(name[i+1])
4. i==0 || (i>0 && isspace(name[i-1]))
5. isalpha(name[i]) && isspace(name[i+1]) && i+1 < strlen(name)

**2. Consider the following code, for all integer values for “a” and “b” being greater than zero.**

int recursive\_function(int a, int b )

{

 if(b==0)

 return 1;

 if(b == 1)

 return a;

 else

 return a \* recursive\_function(a,b-1);

}

**This recursive function …**

1. Computes the factorial “b” multiplied by “a”.
2. Computes “a” to the power of “b”.
3. Computes the factorial of “a” multiplied by “b”.
4. Computes “b” to the power “a”.
5. None of the above.

**3. Consider a function of the following prototype**

void fun (int x, int \*z , int y);

**Assuming that a, b, and c are int variable in the main function. Which of the following will be a correct way to call and use “*fun”* function from the main :**

1. fun(7,&a,b);
2. b = fun(a, \*c, a);
3. c = fun (a,&b,c);
4. c = fun(2,8,1);
5. fun(a\*,b,\*c);

**4. Consider the following code**

int A[10];

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

 printf(“%d”,A[i]);

**The program will print:**

1. The program will print all zeros.
2. All the numbers from 0 to 10.
3. All the numbers from 0 to 9.
4. The program will print garbage values that are meaningless
5. The program will not compile or run because it has an error.

**5. What will be the value of counter after the execution of the while loop**

int x = 3, counter = 0;

while ((x-1))

{

 ++counter;

 x--;

}

1. 0
2. 1
3. 2
4. 3
5. 4

**6. Consider the following recursive function**

int mystery(int a, int b){

 if (b==1)

 return a;

 else

 return a + mystery(a,b-1);

}

**What is the value of mystery(3,3)?**

1. 9
2. 6
3. 3
4. 12
5. the program generates a run time error (infinite recursion)

**Consider the following:**

int array1[10] = {0,1,3,5,6,7,8,22,29,31};

**7. Which of the following statement is true about binary and linear search when trying to find the number 1 in array1.**

1. Binary search will find “1” in fewer iterations (faster).
2. They will find it at exactly the same number of iterations.
3. Binary search cannot be applied to this array.
4. Linear search will find “1” in fewer iterations (faster).
5. None of the above.

**8. Consider the following line of code**

int A[5] = {1};

The content of A will be:

1. All zeros {0,0,0,0,0}
2. One followed by zeros {1,0,0,0,0}
3. All ones {1,1,1,1,1}
4. One followed by garbage values that are meaningless.
5. The program will give a compile error.

**9. What will be printed by the following code?**

int i;

for(i = 1; i <=6 ; i++) {

 if(i<3)

 continue;

 else if (i > 4)

 break;

 printf("%d ",i);

}

1. 1 2
2. 5 6
3. 3 4
4. 1 2 3 4
5. No value printed

**10. Consider the following code**

int i,j;

int ctr = 0;

int myArray[2][3];

for (i=0; i<3; i++)

 for (j=0; j<2; j++)

 {

 myArray[j][i] = ctr;

 ++ctr;

 }

**What is the value of myArray[1][2]**

1. 1
2. 2
3. 3
4. 4
5. 5

**11. Which of the following is an incorrect way to create a 2D array**

1. int A[3][] = {{1,2,3},{4,5,6},{9,8,7}};
2. int A[3][3] = {{1,1,1},{1,1,1},{1,1,1}};
3. int A[][4] = {{1,2,3,4}};
4. int A[10][10];
5. int A[3][3] = {1};

**12. What will be the output of the following program**

char word[50] = "This exam is interesting"; int i;

 for(i=1 ; i< strlen(word) ; i++)

 if(isspace(word [i-1]) && word [i] == 'i')

 printf("%c ", word [i]);

1. i
2. i i
3. s
4. i i i i
5. The program will give an error

# Question 3 (40pts):

**1. (10pts) Complete the following recursive binary search function so that it returns the index of the array element when found or -1 if not found.**

int BinarySearch(double data[],double key,int start, int end, int size){

 if(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

 return -1; //the key was not found in the array

 int middle = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ;

 if( array[middle] == key )

 return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 else

 if(array[middle] < key)

 return BinarySearch(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

 else if(array[middle] > key)

 return BinarySearch(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

 }

**2. (6pts) Complete the program function *FixMe*() given below. Its purpose is to change any digit found in its string parameter to '$'. For example, if the string parameter has the value "ICS 103 IS FUN", then after executing the function it will have the value "ICS $$$ IS FUN"**

**In addition, answer why we should remove the return statement below**

void FixMe(char[] A){

 int i;

 for(i=0;\_\_\_\_\_\_\_\_\_\_\_;i++)

 if(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

 A[i] ='$';

return A;// This statement should be removed because \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

**3. (6pts ) Complete the following function so that it returns the first index within the array of strings “*sentences*”, that contain the string “*key*”. If the “*key*” was not found in any of the “*sentences*”, -1 should be returned. For example, the word “Today is Sunday” contains the string “Sun”.**

int find\_key (char sentences[][LENGTH], char key[LENGTH], int size)

{

 int i,j;

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

 {

 if(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

 {

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 }

 }

 return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ;

}

**4. (6pts) Complete the following function so that it swaps its two arguments. The two arguments for this function are strings.**

void swap\_strings(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) {

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

**5. (6 pts) Complete the following function so that it finds the sum of each column of its argument (2D array) and returns the result as a 1-D array. Assume the type is double.**

void sum\_columns(int A[][COLS] , int sum\_cols[], int rows, int cols) {

 int i,j;

for(i=0; \_\_\_\_\_\_\_\_\_\_\_\_; i++)

{

 for(j=0 ; \_\_\_\_\_\_\_\_\_\_\_\_\_ ; j++)

 {

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 }

}

}

**6.** **(6pts) Complete the following code fragment so that it will find how many capital letters are in the list of strings shown in the initialization**.

int i,j,count;

char text[4][80] = {"THis is ics 103","Final EXam","TErm 102","Good Luck"};

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

for(i=0;\_\_\_\_\_\_\_\_\_\_\_\_\_;i++) {

 for(j=0; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ;j++)

 if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; //print the result