KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

Information and Computer Science Department

2013 Spring Semester (Term 122)

ICS103 Computer Programming in C (2-3-3)

**Final Exam**

May 25, 2013

120 Minutes

|  |  |
| --- | --- |
| Exam Code | **001** |
| Student Name |  |
| KFUPM ID |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Class Section | ADIL AL-SUHAIM  | 02 □ SM 07am | 09 □ SM 1:10pm | 25 □ UT 1:10pm |
| AHMAD AKRAM  | 12 □ UT 07am | 15 □ UT 08am | 21 □ UT 11am |
| AHMAD AL-HERZ  | 01 □ SM 07am | 13 □ UT 07am | 16 □ UT 08am |
| RAFI UL HASAN  | 19 □ UT 9am | 20 □ UT 11am | 24 □ UT 1:10pm |
| S ARAFAT  | 18 □ UT 9am | 22 □ UT 11am | 23 □ UT 1:10pm |
| MOHAMMAD BALAH  | 11 □ UT 07am | 14 □ UT 08am | 17 □ UT 09am |
| EMAD RAMADAN  | 06 □ UM 08am | 07 □ SM 11am |  |
| NASIR AL-DARWISH  | 03 □ UM 07am | 05 □ UM 08am |  |
| ABDULLA AL-SUKAIRY  | 04 □ UM 08am |  |  |
| HAZIM SELMI  | 10 □ SM 1:10pm |  |  |
| M. AL-MULHEM  | 08 □ SM 11am |  |  |
|  |  |  |  |

**IMPORTANT NOTES**

* Fill-in your information on the answer sheet.
* **Answer all fifty (50) questions.**
* **Mark your answers on the answer sheet.**
* **The answer sheet is the only one that will be graded.**
* Do NOT start the exam until you are instructed to do so.
* This is a closed material exam. So, remove any relevant material.
* Calculators are NOT allowed. If you have one, put it on the floor.
* Mobile phones are NOT allowed. If you have one, switch it off NOW.

1) How many times will the following code fragment print “hello”?

 for (i = 2; i<100; i = i\*i)

 printf("hello");

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

2) Assume a and b are int variables with a = 6 and b = 3. Which of the following is true?

1. a > 3 && b > 12
2. b \* b > a + b && a/b > a % b
3. a - b > a % b
4. none of the above

3) Consider the rand()function used in the lab, how to generate a random number between 2 and 40 inclusive?

1. rand() % 41;
2. 2 + rand() % 39;
3. 2 + rand() % 40;
4. 1 + rand() % 42;

4) Which of the following is true about 1-D arrays?

1. An array can contain data items of different data types.
2. An array size can be changed after declaration.
3. A subscript of an array can be an integer expression.
4. The subscript for the last element of an array is the array size.

5) Given the following declaration

 int b[3][2] ={{1, 2}, {3}, {4}};

 What is the value of b[1][1]?

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

6) Suppose a string str is declared and initialized as follows:

 char str [40]= "ABCDEFGHI";

Which of the following loops will change str to "aaaaEFGHI"

|  |  |
| --- | --- |
| A. for(i=0; i<strlen(str); i++) if(isalpha(str[i])) str[i]=’a’; | B. for(i=0; i<strlen(str); i++) if(str[i]==’A’||’B’||’C’||’D’) str[i] = ’a’; |
| C. for(i=0; i<strlen(str); i++) if(str[i]>=’A’||str[i]<= ‘D’) str[i] = ’a’; | D. for(i=0; i<strlen(str); i++) if(str[i]>=’A’&&str[i]<=‘D’) str[i]=’a’; |

7) The following function, fun, is defined using a loop as follows:

int fun(int n) {

int i, sum=0;

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

 sum = sum + i;

return sum;

}

Which one of the following is the correct **recursive** version of the above function?

|  |  |
| --- | --- |
| A. int fun(int n) { return n + fun(n-1); } | B. int fun(int n) { if(n==1) return 1; else return n + fun(n-1);} |
| C. int fun(int n) { if(n>=1) return n + fun(n-1);} | D. int fun(int n) { if(n==1) return 1; else  return fun(n) + fun(n-1);} |

**The next 3 questions are based on the following code fragment:**

The function reverse receives a string and reverses it. For example if the received string is “ABCD” it will become “DCBA”.

The following is an incomplete version of the reverse function:

void reverse (char st[]) {

 int i, k;

 char temp;

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // Statement 1

 for(i=0; i < k/2; i++) {

 temp = st[k-i-1];

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // Statement 2

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_// Statement 3

}}

Fill in the three missing parts (statement 1, statement 2 and statement 3), so that it reverses the string properly.

8) Statement 1 should be:

1. k = strsize(st);
2. k = stringlength(st);
3. k = strlen(st[]);
4. k = strlen(st);

9) Statement 2 should be:

1. st[i] = temp;
2. st[k-i-1] = temp;
3. st[k-i-1] = st[i];
4. st[i] = st[k-i-1];

10) Statement 3 should be:

1. st[i] = temp;
2. st[k-i-1] = temp;
3. st[k-i-1] = st[i];
4. temp = st[i];

11) What will be the output of the following code fragment:

int x[3][3]={{1,2,3},{4,5,6},{7,8,9}}, i, j;

for(i=2; i>=0; i--)

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

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

1. 3 2 6
2. 7 8 4
3. 9 3 1
4. 7 5 3

**The next 2 questions are based on the following incomplete code fragment:**

int x[3][4]={{2,20,-3,1},{4,-11,13,25},{15,42,6,-5}},max,i;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_// Statement 1

for(i=1; i<3; i++)

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_// Statement 2

 max = x[i][3];

The objective of the above code is to find the maximum value in the last column of array “x” and save it in variable max. What should be the missing 2 statements?

12) Statement 1 should be:

1. max = x[0][0];
2. max = x[0][4];
3. max = x[2][0];
4. max = x[0][3];

13) Statement 2 should be:

1. if(x[i][4] > x[i-1][4])
2. if(x[i][3] > x[i-1][3])
3. if(x[i][3] > max)
4. if(x[i][4] > max)

**The next 2 questions are based on the following code fragment:**

int x = 5, y = 3;

int \*p1, \*p2;

p1 = &x;

p2 = &y;

\*p1 = \*p1 + \*p2;

y = x + y;

x = \*p1 + \*p2;

14) What is the final value of x?

1. 8
2. 19
3. 12
4. 11

15) What is the final value of y?

1. 11
2. 12
3. 8
4. 19

**The next 2 questions are based on the following code fragment:**

int A[6] = {11,12,19,27,33,42};

 int i, count=0;

 for (i=0; i<6; i++){

 if (A[i] % 3 == 0) {

 printf("%d ", A[i]);

 if ((A[i] % 6) ==0 )

 continue;

 count++;

 }}

16) What is the output after executing the above code

1. 12 19 27 33 42
2. 12 27 33 42
3. 12 33 42
4. 12 27 33

17) What will be the value of the variable count after the loop

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

18) What is the output of the following C code fragment?

int i, j,count = 0;

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

 count++;

 for(j=1; j<=4; j=j+1)

 count++;

}

printf("%d", count);

A. 16

B. 12

C. 10

D. 15

19) Consider the following code fragment

char a[]="ABCfg123%#@}";

int i, count=0;

for(i=0; i<12; i++) {

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // if statement

count++;}

Which of the following is the right **if statement** so that the value of the variable count becomes 4 at the end of the loop.

1. if(isalnum(a[i]))
2. if(isalpha(a[i]))
3. if(isdigit(a[i]))
4. if(!isalnum(a[i]))

20) Assume the following declaration and missing function call

int x[4][10];

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_// call to function** wow

The function wow has the following prototype:

void wow (int a[ ][10], int rows, int cols);

Which one of the following is the correct call to the function wow**?**

1. wow (x[][10], 4, 10);
2. void wow (x[4][10], 4,10);
3. wow ( &x,4,10);
4. wow (x,4,10);

21) Consider the following incomplete code fragment:

int x[3]={2,5,9};

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//** call the function fun

printf("%d %d %d\n", x[0],x[1],x[2]);

The function fun is defined as follows:

void fun (int \*a, int b, int \*c) {

 \*a = \*a + 2;

 b = b - 1;

 \*c = \*c + 1;

}

Which of the following is the correct call for the function fun to generate: 2 6 11 as the output?

1. fun(&x[1], x[0], &x[2]);
2. fun(&x[2], x[0], &x[1]);
3. fun(&x[0], x[1], &x[2]);
4. fun(x, x[0], x);

22) Which code fragment will assign the following values to array x?

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 1 | 1 |
| 2 | 0 | 1 | 1 |
| 2 | 2 | 0 | 1 |

|  |  |
| --- | --- |
| I. int i,j, x[3][4]={0}; for (i=0; i<3; i++) { for (j=0; j<4; j++) { if(i > j) x[i][j]=2; else x[i][j]=1; }} | II. int i,j, x[3][4]; for (i=0; i<3; i++) { for (j=0; j<4; j++) { if(i > j) x[i][j]=2; else if (i < j) x[i][j]=1; }} |
| III. int i,j, x[3][4]={0}; for (i=0; i<3; i++) { for (j=0; j<4; j++) { if(i > j) x[i][j]=2; else if (i < j)  x[i][j]=1; }} | IV. int i,j, x[3][4]={0}; for (i=0; i<3; i++) { for (j=0; j<4; j++) { if(i >= j) x[i][j]=2; else  x[i][j]=1; }} |

1. I
2. II
3. III
4. IV

23) Consider the **selection sort** function. Select the correct order of the elements of the array after each pass to sort the following array: {8,6,4,5,7}

1. {4,6,5,7,8} {4,6,5,7,8} {4,5,6,8,7} {4,5,6,7,8}
2. {4,8,6,5,7} {4,5,8,6,7} {4,7,5,6,8} {4,5,6,7,8}
3. {4,6,8,5,7} {4,5,8,6,7} {4,5,6,8,7} {4,5,6,7,8}
4. {4,6,8,5,7} {4,6,8,5,7} {6,4,5,8,7} {4,5,6,7,8}

24) Consider the following code fragment:

for(i=1; i<=6; i=i+2)

 for(j=5; j>=0; j=j-2)

 printf("ICS 103\n");

How many times is “ICS 103” printed?

1. 12
2. 9
3. 6
4. 18

25) What is the output generated by the following call?

printf("%d ",mystery(1,5));

The function mystery is defined as follows:

int mystery(int a, int b) {

 if(a==b)

 return 1;

 else {

 printf("%d ",b-a);

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

 }}

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

26) Which one of the following is the correct code that reads 10 values and assigns them to the elements of the array y?

**Note**: y is declared as an array of type int and has size 10.

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

 scanf(“%d”,&y[10]);

B. for(i=1; i<=10; i++)

 scanf(“%d”,&y[i]);

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

 scanf(“%d”,&y[i]);

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

 scanf(“%d”,y[i]);

27) What is the output of the following code fragment?

int x[3][3]={{1,2,3},{4,5,6},{7,8,9}},i;

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

 printf("%d ",x[2-i][i]);

1. 7 5 3
2. 1 5 9
3. 9 5 1
4. 3 5 7

28) What is the output of the following code fragment?

int check=2;

switch(check){

  case 1: printf("The");

  case 2: printf("#ICS103");

  case 3: printf("#Final");

  default: printf("#Exam");

}

1. #ICS103#Final#Exam
2. #ICS103#Final
3. #ICS103
4. #ICS103

 #Final

29) What is the output of the following code fragment?

int w, i=1;

printf("%d ",i++);

w = 2\*++i;

printf("%d %d",i,w);

A. 1 3 6
B. 2 2 4
C. 1 2 4

D. 2 3 6

**The next 2 questions are based on the following code fragment:**

scanf("%d%d",&x,&y);

if(x>10)

 if(y<20)

 if (x > y)

 printf("1");

 else

 printf("2");

 else

 printf("3");

else

 printf("4");

30) What is the output if the user has typed **15 22** as input

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

31) What is the output if the user has typed **12 12** as input

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

32) What is the output after executing the following code fragment?

 char line[10]="ABD123" ;

 tolower(line[1]);

 tolower(line[3]);

 puts(line);

1. aBd123
2. AbD023
3. ABD123
4. AbD123

33) Consider the following declaration and initialization:

char cities[3][15]={"Dammam","Khobar","Taif"};

What is the correct way to display the length of string “Khobar” ?

1. printf("%d ",strlen(cities[2][6]));
2. printf("%d ",strlen(cities[1]));
3. printf("%d ",strlen(cities[1][15]));
4. printf("%d ",strlen(cities[2]));

34) What is the output of the code fragment shown below?

int x[4]={3,4,-2,1}, i=-1, sum=0;

do{

 i++;

 sum = sum + x[i];

} while(x[i] > 0);

 printf("%d ",sum);

1. 7
2. 8
3. 5
4. 6

35) What will be printed by the following statement?

printf("%d ",strcmp("am","abc"));

1. Positive Value
2. Negative Value
3. 0
4. None of the above

36) Consider **the bubble sort** function. Select the correct order of the elements of the array after each pass to sort the following array: {8,7,5,9,2}

1. {2,8,7,5,9} {2,5,8,7,9} {2,5,7,8,9} {2,5,7,8,9}
2. {2,8,7,5,9} {2,5,7,9,8} {2,5,7,9,8} {2,5,7,8,9}
3. {7,5,8,2,9} {5,7,8,2,9} {2,5,8,7,9} {2,5,7,8,9}
4. {7,5,8,2,9} {5,7,2,8,9} {5,2,7,8,9} {2,5,7,8,9}

**The next 5 questions are based on the following code**

#include <stdio.h>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; // Statement 1 (prototype)**

int main() {

 int x[3]={1,2,3},y[3]={4,-1,6},prod;

 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_// Statement 2 (function call)**

 printf("scalar product of x and y= %d\n",prod);

 return 0;

}

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ { //statement 1 (function header)**

 int i,p;

 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // Statement 3 (Initialization)**

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

 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //Statement 4 (Calculation)**

 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // Statement 5**

}

The above code is made of the main and scalarp functions. scalarp receives two integer arrays of size m, and returns the scalar product of the two arrays.

The scalar product of two arrays is defined as:

product=x[0]\*y[0]+x[1]\*y[1]+……+x[m-1]\*y[m-1]

37) Statement 1 should be:

1. void scalarp (int x[], int y[], int m, int p)
2. int scalarp (int x[], int y[], int m)
3. void scalarp (int x[], int y[], int m, int &p)
4. int scalarp (int x[m], int y[m])

38) Statement 2 should be:

1. prod = scalarp (x,y,3);
2. void scalarp (x[], y[], 3, prod);
3. scalarp (x[3], y[3],prod);
4. prod = int scalarp (x,y,3);

39) Statement 3 should be:

1. p = 1;
2. p = x[0] \* y[0];
3. p = 0;
4. p = x[3] \* y[3];

40) Statement 4 should be:

1. p = x[3] \* y[3];
2. p = p \* x[i] \* y[i];
3. p = x[i] \* y[i];
4. p = p + x[i] \* y[i];

41) Statement 5 should be:

1. printf("%d ",p);
2. printf("%d ",scalarp);
3. return p;
4. return scalarp;

**The next 2 questions are based on the following code fragment:**

char \*token, delim[] = "!";

char str[] ="This!is!ics!103";

int count=0, total=0;

token = strtok(str, delim);

while(token != NULL){

 count++;

 total = total + strlen(token);

 token = strtok(NULL, delim);

}

42) What will be the value of count after the while loop?

1. 4
2. 5
3. 6
4. 7

43) What will be the value of total after the while loop?

1. 15
2. 14
3. 13
4. 12

**The next 2 questions are based on the following code fragment:**

char line[20],text[3][7]={"ICS103","122","Great!"};

int i, j, count=0, k=0;

for(i=2;i>=0;i--)

 for(j=0; text[i][j] != '\0'; j++)

 if(isalpha(text[i][j])){

 line[k] = text[i][j];

 k++;

 }

 else

 count++;

line[k]='\0';

puts(line);

44) What is the output of the above code fragment?

1. Great!ICS
2. Great!122ICS103
3. GreatICS
4. !122103

45) What will be the final value of count after executing the code?

1. 6
2. 11
3. 7
4. 9

46) Which of the following codes will print the pattern shown below?

\*\*\*\*

\*\*\*

\*\*

\*

|  |  |
| --- | --- |
| A. for(i=4; i>=1; i--) for(m=1- printf()1;i--)owing codes prints the pattern shown below: as:; m<=i; m++) printf("\*"); printf("\n"); | B. for(i=4; i>=1; i--) {for(m=4; m>=1; m--) printf("\*");printf("\n");} |
| C. for(i=1; i<=4; i++){ for(m=1; m<=i; m++) printf("\*"); printf("\n");} | D. for(i=1; i<=4; i++){ for(m=1; m<=5-i; m++) printf("\*"); printf("\n");} |

47) Consider the following code fragment

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

int i,j,sum;

for(j=0; j< 3;j++) {

 sum = 0;

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

 if (i !=j)

 sum = sum + A[i][j]; }

 printf("%d ", sum);

}

What is the output of the above code?

1. 11 10 9
2. 5 10 15
3. 9 11 10
4. 11 10 5

48) Consider the following array:

int x[7]={7, 4, 10, 8, 4, 9, 5};

The call to the **linear search** function covered in class with array x and target value of 4 will return the following value

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

49) Consider the following array:

 int x[7]={10, 15, 19, 25, 32, 40, 62};

and consider searching for the value **12** by the **recursive binary search** covered in the class and having the following prototype:

int binarySearch (int x[], int first, int last, int target);

the values of **first** and **last** in the second call to binary search are:

1. first = 0, last = 3
2. first = 1, last = 4
3. first = 4, last = 6
4. first = 0, last = 2

50) Consider the following array:

int x[7]={10, 13, 23, 25, 30, 35, 50};

and consider searching for the value **27** by the **recursive binary search** covered in the class. The first 2 values of the array compared to the **target** value are:

1. 25 and 35
2. 25 and 30
3. 50 and 35
4. 25 and 50