

Information and Computer Science Department

Fall Semester 141

ICS 103 – Computer Programming in C

Midterm Exam

Saturday, November 08, 2014

Duration: 120 minutes

|  |  |
| --- | --- |
| Name: | KEY |

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

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

|  |  |
| --- | --- |
| Instructor: |  |

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

 AAAAAAAAAAAAAAAAAAQuestion # 1 [15 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 next page.

Q. 1:

Which of the following tasks is typically not done by an operating system?

A. Communicate with computer user,

B. Manage memory,

C. Collect input/Display output,

D. Do simple word-processing.

Q. 2:

Which of the following is the correct order of the use of various application software for entering, translating and running a complete ‘C’ program?

A. word processor (editor), loader, compiler, linker

B. word processor (editor), compiler, linker, loader

C. loader, linker, compiler, word processor (editor)

D. word processor (editor), compiler, loader, linker

Q. 3:

'#include' used in the beginning of a 'C' source file is:

A. a preprocessor directive

B. a predefined function

C. a constant definition

D. a function prototype

Q. 4:

Which of the following is not a ‘C’ reserved word?

A. double

B. return

C. printf

D. if

Q. 5:

Which of the following can be a valid C identifier?

A. three\_stars

B. 7up

C. myname@apple

D. miles per kilometre

Q. 6:

Which of the following is not correct regarding ‘C’ comments?

A. comments help us to understand the program, but are ignored by the compiler

B. multi-line comments begin with a ‘/\*’ and end with a ‘\*/’

C. one-line comments begin with a ‘//’

D. every C program must have comments in order to compile and execute correctly

Q. 7:

Which of the following is the correct 'C' code for the formula: *root1* =

A. root1 = (-b + sqrt(b\*b - 4\*a\*c)) / 2\*a

B. root1 = (-b + sqrt(b\*b - 4\*a\*c) / (2\*a))

C. root1 = (-b + sqrt(b\*b - 4\*a\*c)) / (2\*a)

D. root1 = -b + sqrt(b\*b - 4\*a\*c)/(2\*a)

Q. 8:

Which of the following is a correct function prototype for a void function with two integer arguments ?

A. void myfunction(int a, b);

B. myfunction void(int a, b);

C. void myfunction( int a, int b);

D. myfunction void( int a, int b);

Q. 9:

Which of the following is the correct order of evaluation for the C operators in the expression shown below?

 5 == 3 > 6 + 11 \* 7.5 || 1 && flag

A. ==, >, +, \*, ||, &&

B. \*, +, >, ==, &&, ||

C. +, \*, ==, >, &&, ||

D. \*, +, ==, >, &&, ||

Q. 10:

Which of the following is not a correct increment operation?

A. counter = counter + 1;

B. counter++;

C. counter += 1;

D. counter + 1;

Q. 11:

Which of the following is a correct implementation of the condition

 10 ≤ m ≤ 14?

A. (m >= 10 || m <= 14)

B. (10 <= m <= 14)

C. (m >= 10 && <= 14)

D. (m >= 10 && m <= 14)

Q. 12:

The switch statement cannot be used on which of the following variable types:

A. int

B. char

C. double

D. short

Q. 13:

What will be the output of the following C code fragment?

 double x = 3.456;

 printf("%4.1f\*%5.2f\*%.4f\n", x, x, x);

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A. |  | 3 |  . | 4 | \* |  | 3 | . | 4 | 5 | \* | 3 | . | 4 | 5 | 6 | 0 |
| B. |  | 3 |  . | 5 | \* |  | 3 | . | 4 | 6 | \* | 3 | . | 4 | 5 | 6 | 0 |
| C. |  | 3 |  . | 5 | \* | 3 | . | 4 | 6 | \* | 3 | . | 4 | 5 | 6 | 0 |  |
| D. |  | 3 |  . |  4 | \* |  3 |  . | 4 |  5 | \* | 3 |  . |  4 | 5 | 6 | 0 |   |

Q. 14:

What will be the output of the following C code fragment?

 int x = 345

 printf("%4d\*%5d\*%1d\n", x, x, x);

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A. |  | 3 | 4 | 5 |  \* |  |  | 3 | 4 | 5 |  \* | 3 | 4 | 5 |  |  |
| B. |  | 3 | 4 | 5 |  \* |  |  | 3 | 4 | 5 |  \* | 3 |  |  |  |  |
| C. |   | 3 | 4 | 5 |   |  \* |  |  | 3 | 4 |  5 |  |  \* | 3 |  |  |
| D. | 3 | 4 | 5 |  |  \* |  3 | 4 | 5 |  |   |  \* | 3 | 4 | 5 |  |  |

Q. 15:

What will be the output of the following code fragment?

 printf("%f\n ",((int) (3.141592\*1000))/1000.0);

A. 3.141592

B. 3.141000

C. 3.142000

D. 3.000000

FORM AAAAAAAAAAAAAAAAAAAAAA

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

Question # 2 [15 points] FORM AAAAAAAAAAAAAAAAAAAAAAAAAA

(a) What will be the outputs of the following programs?

| Program | Output |
| --- | --- |
| #include <stdio.h> int test(int i){ int x = 0, j; for (j =1; j<3; j++){ x = x + i%10; i = i/10; printf("%d\n",x); } return 5;} int main() { printf("%d\n", test(24));  return 0;} | 3 points465 |
| #include <stdio.h>int main(void){ int i, count; count=0; for(i=1;i<=8; i++) if(i%2==0)  count++;  printf("%d", count); return 0;} | 2 points4 |
| #include <stdio.h>int main(void){ int x= 7, y = 4,z;  printf("%d\n",x--); z= --x \* y++; printf("%d \n", z); return 0;} | 2 points720 |
| #include <stdio.h>void a(void);void b(void);int main(void){ a(); b(); return 0;}void b(void){ printf("Welcome\n");}void a(void){ printf("Hello\n");} | 2 pointsHelloWelcome |

FORM AAAAAAAAAAAAAAAAAAAAAAAAAAAA

(b) Consider the following programs. What will be the output for each of the input values typed by the user?

(b-1)

#include <stdio.h> // 3 points

int main() {

 int x;

 printf("Enter a value for x >");

 scanf("%d",&x);

 if(x >= 6 ) {

 if(x < 10){

|  |  |
| --- | --- |
| Value of x typed by user | Program output |
| 15 | D |
| 7 | C |
| 3 | F |

 if(x > 7)

 printf("A");

 else

 printf("C");

 }else{

 if ( x > 12)

 printf("D");

 else

 printf("B");

 }

 }

 else

 printf("F");

 return 0;

}

(b-2)

#include <stdio.h> // 3 points

int main() {

 int y;

 scanf("%d",&y);

|  |  |
| --- | --- |
| Value of y typed by user |  Program output  |
| 7 | 10 |
| 2 | 9 |
| 4 | 8 |

 switch(y){

 case 7: y=y+1;

 case 3: y=y+2;

 break;

 case 2: y=y+3;

 default: y=y+4;

 }

 printf("%d\n",y);

 return 0;

}Question # 3 [15 points]

Assume that a bookstore gives 5% discount in case that your total purchasing amount is more than SR100; otherwise no discount will be given. Consider the *problem* of computing the final net price given the total purchasing amount, state and apply the first 4 steps of the software development method to solve it.

1. Problem

A bookstore gives 5% discount in case that your total purchasing amount is more than SR100; otherwise no discount will be given. Consider the *problem* of computing the final net price given the total purchasing amount

1. Analysis
* Input(s): total purchasing amount in SR
* Output(s): final net price
* Formula (conditional): final net price = total purchasing amount \* discount rate
1. Design

Algorithm

1. Get the total purchasing amount in SR
2. Apply the 5% discount rate if the amount is greater than 100
3. Display the final net price
4. Implementation

#include <stdio.h>

#define DISCOUNT 0.05

int main(void) {

 double final\_price;

 printf(“Please enter the total purchasing amount in SR >”);

 scanf(“%lf”,&final\_price);

 if (final\_price>100)

 final\_price = final\_price \*(1 - DISCOUNT ) ;

 printf(“The final net price is SR %.2f”, final\_price);

}

Question # 4 [15 points]

Write a C program that estimates the temperature T (in °C) in a freezer in case of power failure. The temperature is given by the following formula:

$$T=\frac{4\sqrt{t}}{t+2}-20$$

where t is the time (in minutes) since the power failure.

Your program should prompt the user to enter how long it has been since the start of the power failure in whole hours and minutes. Note that you will need to convert the elapsed time into minutes.

For example, if the user typed 2 30

as input, which means 2 hours and 30 minutes. You program would need to convert this to minutes to get 150 minutes.

#include <stdio.h>

#include <math.h>

int main(void) {

 int hours, minutes, elapsed\_time;

double temperature;

 printf(“Enter time since the power failure in whole hours and minutes”);

 scanf(“%d%d”,&hours,&minutes);

 elapsed\_time = 60\*hours + minutes;

 temperature = (4\* sqrt(elapsed\_time))/( elapsed\_time+2) -20;

 printf(“The estimated temperature is %f”, temperature);

 return 0;

}

Question # 5 [20 points]

Complete the following C program by filling the 3 blanks and writing the function definition of function operation.

This function takes as inputs: two real numbers x, y and an operator as a character. The operator is one of the following: ‘+’, ‘-’, ‘/’, or ‘\*’.

The function will then return (not print) the result of the operation. This result should be displayed in the main function as shown in the sample runs.

Your function must use a switch statement to find the result of the operation.

 



#include <stdio.h>

double operation(double, double, char); // function prototype

int main(){

 double a,b,result;

 char op;

 printf("Enter operation, a op b : ");

 scanf("%lf %c %lf",&a,&op,&b);

 if(op=='/' && b==0 )

 printf("Error cannot divide by 0\n");

 else if ( op == '\*' || op == '+' || op == '-' || op == '/') {

 result = operation(a,b,op);

 printf("%.2f %c %.2f = %.2f \n",a,op,b,result);

 }

 else

 printf("%c is unknown operation\n", op);

 return 0;}

// missing function definition should be written below this line

double operation(double a, double b, char op){

 double result;

 switch(op){

 case '\*': result=a\*b;

 break;

 case '+': result=a+b;

 break;

 case '-': result=a-b;

 break;

 case '/': result=a/b;

 }

 return result;}

// other solution using if-else-if

double operation(double a, double b, char op){

 double result;

 if(op=='\*')

 result=a\*b;

 else if(op=='+')

 result=a+b;

 else if(op=='-')

 result=a-b;

 else if(op=='/')

 result=a/b;

 return result;

}

Question # 6 [20 points]

The value of the mathematical constant e can be expressed using the Tailor series as follows:

$$e^{x}=1+\frac{x}{1!}+\frac{x^{2}}{2!}+\frac{x^{3}}{3!}+…, -\infty <x<\infty $$

Where $n!=1×2×3×4×5×…….×n$

Write a C program that will prompt the user for a value of x. Then, it will compute the first 21 terms of Tailor expansion for the input value of x (x must be a double value).

(Hint: the 2nd term of above equation can be re-written as $\frac{x}{1}$

 The 3rd term of above equation can be re-written as $\frac{x}{1}.\frac{x}{2}$

 The 3rd term of above equation can be re-written as $\frac{x}{1}.\frac{x}{2}.\frac{x}{3}$ and so on )

#include<stdio.h>

int main(){

 double x,term,result;

 int n;

 printf("Enter a value for x:");

 scanf("%lf",&x);

 term = 1.0;

 result = 1.0;

 n=1;

 while(n<=20){

 term = term\*x/n;

 result = result+term;

 n = n+1;

 }

 printf("The result is :: %f", result);

 return (0);

}