

16.070 Exam #1

March 4, 2002

Name: _____

- 1) (5 points) Circle the correct answer. An algorithm refers to
- a) **A step-by-step solution to solve a specific problem.**
 - b) A collection of instructions that the computer can understand.
 - c) A code that allows us to type in text materials.
 - d) Stepwise refinement.
 - e) A set of math equations to derive the problem solution.
- 2) (5 points) Circle the correct answer. The execution of a program begins with the main function.

T F

- 3) (5 points) Circle the correct answer. A computer program is the implementation of an algorithm.

T F

- 1) (10 points) Write a **C expression**, including an assignment operator, to compute the length of the third side "a" of a triangle, given the length of the first two sides "b" and "c", the angle between them "θ", and the following algebraic expression:

$$a^2 = b^2 + c^2 - 2bc \cos \theta$$

Assume that floating point variables of type "double" have been defined for a, b, c, and d (for θ), and that d is provided in radians.

SOLUTION:

a = sqrt ((b*b) + (c*c) - (2*b*c* cos(d)))

SOLUTION:

- 5) (9 points) The following function will generate several compiler warnings and errors. Identify 3 places where warnings or errors occur and a brief reason for each. Line numbers are included for your convenience.

```
1 #include <stdio.h>
2
3 int ExamFunction(int Size) {
4     const float TWICESIZE = Size*2;
5     int my_array[TWICESIZE]={0};
6
7     printf("my array has %d bytes", sizeof(my_array));
8     return TWICESIZE;
9 } /* end ExamFunction */
```

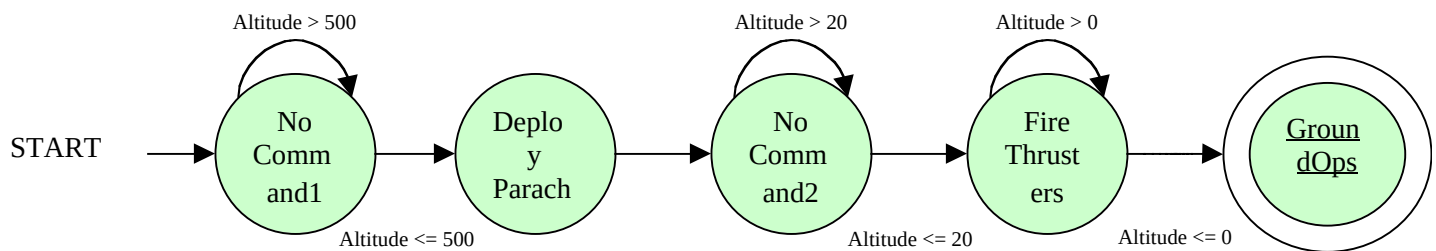
Warnings/Error:

Code Line #	Description
4	int assigned to float
5	index is a float
5	index is a const vs #define or literal constant
8	float return from int function

6) (13 points) Draw the following pseudocode algorithm for a Mars lander as a state transition diagram. States are labeled in **bold**.

```
1 SET Altitude = 1000
2 WHILE (Altitude > 500)
3     No Command 1
4 END WHILE
5 Deploy Parachute
5 WHILE (Altitude > 20)
6     No Command 2
7 END WHILE
8 WHILE (Altitude > 0)
9     Fire Thrusters
10 END WHILE
11 GroundOps
```

Solution



- 7) (12 points) Trace the following program, showing all the values of Counter and InnerCntr, and show the output of the program.

```
#include <stdio.h>
#define BASE_LENGTH 3

int main(void) {
    int StrangeInit = 4;
    int Counter=StrangeInit--;
    int InnerCntr = 0;

    for (Counter=0; Counter<BASE_LENGTH; Counter++)
    {
        for (InnerCntr=0; InnerCntr<=Counter; InnerCntr++)
        {
            putchar('*');
        } /* end inner for */
        putchar('\n');
    } /* end outer for */

    return 0;
} /* end main */
```

Variable	Value
Counter	4 0 1 2 3
InnerCntr	0 1 0 1 2 0 1 2 3

- 8) (9 points) The following function will determine the next state in a state transition diagram (not pictured) based on an integer input.

```
1  int ProcessStateChart(int InputVal) {
2      int NextState = 0;
3
4      if ((Input>0)&&(Input<5)) {
5          NextState = 1;
6      }
7      else if ((Input>5)&&(Input<=20)) {
8          NextState = 2;
9      }
10     else {
11         NextState = 3;
12     }
13
14     return NextState;
15 } /* end ProcessStateChart */
```

Statement 10 handles what condition(s)? Answer by writing a new line 10 with an **else if** statement that covers the exact same conditions as the **else** statement above. Assume InputVal can take on any integer value.

ANSWER

```
else if ( (Input<=0) || (Input==5) || (Input>20) ) {
```

- 9) (16 points) **Write a function called "reverse_print" that will print numbers from an array in reverse order on a single line.** In this example, the array will contain the Fibonacci sequence from 0 to 55. The Fibonacci sequence is calculated by the following equation: $f(k) = f(k-1) + f(k-2)$, and the first two elements are 0, 1. For example, the first 8 elements are 0, 1, 1, 2, 3, 5, 8, 13. **Fill in the blanks below**, and write the function to perform the reversal. Assume that the For loop in the main function produces the correctly ordered Fibonacci series. You are given the following code snippet to interface with:

```
#include <stdio.h>
#define SIZE 11

/* Function prototype */
void reverse_print(int array[]);

void main(void)
{
    int arr_int[SIZE]={0}; /* array to hold Fibonacci sequence */
    int counter=1;

    arr_int[1]=1;

    /* Fill in the array with the Fibonacci sequence */
    for (counter=2; counter<SIZE; counter++)
    {
        arr_int[counter]=arr_int[counter-1]+arr_int[counter-2];
    } /* end for loop */

    /* Now call reverse_print function to print out the Fibonacci sequence in reverse order */
    reverse_print(arr_int);
} /*end main*/
```

Write reverse_print function here:

```
Void reverse_print(int my_array[]) {
    Int I = 0;
    For (I=size, I>=0, I--) {
        Printf ("%d, ", my_array[I]);
    } /* end for */
    return;
} /* end reverse_print */
```

10) (16 points) Trace the variables in this program and record the first 13 lines of output in the tables provided. The first two output lines are already filled in to get you started.

```
#include <stdio.h>

#define TRUE 1
#define FALSE 0

#define STATE_MAIN 0
#define STATE_FOO 1
#define STATE_BAR 2
#define STATE_BAZ 3
#define STATE_MIT 4

/* function prototypes */
void do_main(int cycle);
void do_foo(void);
void do_bar(void);
void do_baz(void);
void do_mit(void);

/* global variables */
int g_state = STATE_MAIN;

int main(void) {
    int cycle = 0;
    while(TRUE) { /* loop 'forever' */
        printf("switch\n");
        switch(g_state) {
            case STATE_FOO:
                do_foo();
            case STATE_BAR:
                do_bar();
            case STATE_BAZ:
                do_baz();
                cycle = 1;
                break;
            case STATE_MIT:
                do_mit();
            default:
                do_main(cycle);
        } /* end switch(g_state) */
    } /* end while TRUE */
} /* end main */

void do_main(int cycle) {
    printf("main!\n");

    switch(cycle) {
        case 0:
            g_state = STATE_FOO;
            cycle = 1;
            break;
        case 1:
            g_state = STATE_MIT;
            cycle = 0;
            break;
    } /* end switch(cycle) */
} /* end do_main */

void do_foo(void) {
    printf("foo!\n");
    g_state = STATE_BAZ;
}

void do_bar(void) {
    printf("bar!\n");
    g_state = STATE_MIT;
}

void do_baz(void) {
    printf("baz!\n");
    g_state = STATE_MAIN;
}

void do_mit(void) {
    printf("MIT!\n");
    g_state = STATE_MAIN;
}
```

Variable	Value
G_state	0 1 3 4 0 4 0 4
Main: cycle	0 1
Do_main: cycle	0 1 0 1 0

Output Line #	Program Output
1	switch
2	main!
3	switch
4	foo!
5	bar!
6	baz!
7	switch
8	main!
9	switch
10	MIT!
11	main!
12	switch
13	MIT!