

- This examination is closed-book and closed-notes. You have 50 minutes to complete it. The total number of points is 55.
- Please answer all questions. The exam is printed two-sided, so please make sure you don't miss any questions.
- *Please read through all questions before starting to answer any. It is advisable that you determine which questions you are most comfortable with or that will be easiest for you to answer correctly, and do those first.*
- Please make sure you read and sign the honor code below.

I pledge my honor that I have not violated the honor code during this examination.

For instructor use only:

Question	Points
I	
II	
III	
IV	
V	
Total	

Question I. Short Answer Questions [15 points]

(a) Number Systems:

How is the decimal number 107 represented as an eight-bit binary number? What is the representation in hexadecimal notation of 107? What is two's complement of 107, expressed as an 8-bit binary number?

(b) Performance:

We said in lecture that it is generally thought that 80% of a program's execution time is spent in 20% of its lines of code. Suppose that you have a program for which this is true, and your focus in performance improvement is only on those 20% of the lines of code. If your target is to achieve a 2x improvement in the program's execution time (i.e. to halve the execution time), by how much would you need to improve that portion of the code?

(c) How many input data sets does it take to “statement test” the following code, and how many to “path test” it. Briefly state why. We expect here a number, followed by at most two sentence of explanation in each case.

```
if (condition1)
    statement1;
else
    statement2;

...
if (condition2)
    statement3;
else
    statement4;

...
if (condition3)
    statement 5;
else
    statement 6;
```

(d) What is the value of the expression $(0 < j < 1)$? Does this expression achieve the goal of checking that the value of j is between 0 and 1, exclusively? If so, say why? If not, write an expression that does that. Note that j is not necessarily an integer.

Question II. 10 points

Consider this program:

```
#include <stdio.h>
#include "stack.h"

enum {MAX_LINE_LENGTH = 256};

int man(void)
{
    char line[MAX_LINE_LENGTH];
    Stack_T s;
    s = Stack_new();

    while (fgets(line, MAX_LINE_LENGTH, stdin) != NULL)
        Stack_push(s, line);

    while (! Stack_isEmpty(s)) {
        fputs((char*)Stack_top(s), stdout);
        Stack_pop(s);
    }

    Stack_free(s);

    return 0;
}
```

The program's job is to read lines from stdin until end-of-file, and then write those lines to stdout in reverse order. It uses the fgets() function to read each line from stdin, a Stack object (as defined in precepts) to store the lines, and the fputs() function to write each line to stdout.

For example, if stdin contains these three lines:

Line one.
Line two.
Line three.

then the program should print these three lines to stdout:

Line three.
Line two.
Line one.

However, the program contains a bug. It prints these three lines to stdout:

Line three.
Line three.
Line three.

(a) Describe the bug that causes the incorrect output.

(b) Rewrite the program to eliminate that bug. Your program should be well-styled, but need not include comments.

Question III. (10 points)

What is the output of this program when built with gcc217 and run on the “hats” machines (fez or fedora)?

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

enum {SIZE=10};

char ca3[SIZE];

struct sCos217
{
    char *pcKey;
    int iValue;
    struct sCos217 *psNext;
};

void foo(char ca[SIZE]) {
    printf("(7) %d\n", sizeof(ca));
    printf("(8) %d\n", sizeof(ca3));
    printf("(9) %d\n", strlen(ca3));
}

int main(void) {

    int *pi = (int *)malloc(sizeof(int));
    struct sCos217 *ps = (struct sCos217 *)malloc(sizeof(struct sCos217));
    char ca1[] = "abcd";
    char ca2[] = {'1', '2', '3', '4'};

    printf("(1) %d\n", sizeof(pi));
    printf("(2) %d\n", sizeof(ps));
    printf("(3) %d\n", sizeof(ca1));
    printf("(4) %d\n", strlen(ca1));
    printf("(5) %d\n", sizeof(ca2));
    printf("(6) %d\n", strlen(ca2));
    foo(ca3);

    return 0;
}
```

Question IV. (10 points)

Imagine you are hired by NASA to design a DFA for them. They want to use digital communication (i.e., communication with just 0s and 1s) and they want to make sure they can differentiate their messages from any alien communication. Their requirement is that a) All messages must end with '11'. b) There should not be a code '0010' anywhere in the message. c) The message can be arbitrarily long.

Design a DFA that either accepts or rejects a message based on these requirements. Your DFA should clearly specify the start state by labeling it "S", and any accepting states with a double circle (concentric circles close together). You can assume that the only possible inputs are 1 and 0.

Question V. (10 points)

(a) Consider the following code segment. Does it compile correctly? Does it produce any compiler warnings, and why? Does it produce any compilation errors, and why? What happens if I execute the program?

```
#include <stdio.h>

int main(void) {

    int i;
    int *pi;
    int **ppi;
    i = 3000;

    ppi = &i;
    pi = &ppi;

    printf("i=%d\n", i);
    printf("i=%d\n", *pi);
    printf("i=%d\n", **ppi);

    return 0;

}
```


(b) What is the output of the following program?

```
void foo(int **ppi1, int **ppi2) {  
  
    int *piTemp;  
    piTemp = *ppi2;  
    *ppi2 = *ppi1;  
    *ppi1 = piTemp;  
  
    return;  
  
}  
  
int main(void) {  
  
    int i1 = 10;  
    int i2 = 20;  
    int *pi1 = &i1;  
    int *pi2 = &i2;  
  
    foo(&pi1, &pi2);  
  
    printf("i1=%d\n", i1);  
    printf("i2=%d\n", i2);  
    printf("*pi1=%d\n", *pi1);  
    printf("*pi2=%d\n", *pi2);  
  
    return 0;  
  
}
```