

## CS 246 MIDTERM EXAM

1. You will receive an email from me with your exam at the start of class.
2. You have 80 minutes to complete this exam. You have an additional 15 minutes for electronic submission.
3. All answers should be written on blank (or lined) sheets of paper. Also, answers may be typed or written using a tablet.
4. **SUBMISSION:** send your answers to [gtowell246@cs.brynmawr.edu](mailto:gtowell246@cs.brynmawr.edu). If you typed everything, or wrote on a tablet, just copy and paste or attach the document. For handwritten answers, take photos or scan. Please ensure legibility.
5. Be sure your answers are clearly labeled and that every page has your name
6. This is an open everything exam. The only restriction is that you may not discuss the questions or your answers with anyone.
7. In order to be eligible for as much partial credit as possible, show all of your work for each problem, and clearly indicate your answers. Credit cannot be given for illegible answers.
8. Code that you write should be as close to correct (runnable) as possible. Small syntax errors will not cost you points, but code that is unclear will. You may write your code using VSC; you may compile and test. I do not encourage this (I even discourage), I just note that it is permitted.
9. If you make an assumption, state that assumption clearly.
10. I will be available on zoom for the entire exam period. I suggest you have an open zoom session but with the sound off. If you have a question turn the sound on and ask. I will also watch for chats, but I am notoriously poor at noticing those.
11. Read the entire exam before answering any questions, so you have time to think about the potentially tricky problems.

The following functions may be used (and might be useful) in this exam (or practice):

stdio.h	int putchar(int c);	write a character to standard output
stdio.h	int getchar(void);	read a character from standard input
stdio.h	char *fgets(char *s, int size, FILE *stream);	read a line from stream into s. return NULL on EOF
ctype.h	int toupper(char c); int tolower(char c);	return upper case / lower case of the passed char.
stdlib.h	int rand(void);	returns a pseudo-random integer in the range 0 to RAND_MAX

You may also use any other function defined in stdio.h, stdlib.h, ctype.h and string.h, except as noted in particular questions.

### Problem 1: 10 points

The trim function takes a string and returns that string with all leading and trailing whitespace removed. You may assume that whitespace is either a space (' ') or tab character ('\t'). The prototype for trim is:

```
char* trim(char* target);
```

Implement trim using no functions from string.h, ctype.h, stdio.h or stdlib.h other than those listed above (which you may or may not need). When using your implementation the following will print "No leading whitespace" if target has no leading whitespace.

```
char* result = trim(target);  
if (result==target)  
    printf("NO leading whitespace");
```

### Problem 2: 10 points

What is the output of the following C program?

```
#include <stdio.h>  
int a[][2] = {{2,3},{3,4},{5,6},{7,8}};  
int main(void)  
{  
    printf("%d\n", (int *)a==(int *)a[0]);  
    printf("%d\n", (int *)a==(int *)a[0][0]);  
    int* b = (int *)a[2];  
    int* c = (int *)a;  
    printf("%d\n", b[1] == *(c+2*2+1));  
    printf("%d\n", b[0]=b[1]);  
    printf("%d\n", *(c+6)==*(b+2));  
}
```

### Problem 3: 10 points

```
#include <stdio.h>
int main(void)
{
    char c;
    while (EOF != (c=getchar()))
        printf("%c", c);
}
```

Assume that the above program is stored in a file name `aaa.c`. What does the the following set of commands do?

```
UNIX> gcc -o aaa aaa.c
```

```
UNIX> ./aaa < aaa.c > bbb
```

### Problem 4:10 points

What is the output of the following program?

```
#include <stdio.h>
int main(void)
{
    int i=5;
    int* ip = &i;
    int j = 10;
    int* jp= &j;
    printf("%d %d %d\n", i, j, *jp);
    i++;
    (*jp)++;
    printf("%d %d %d\n", i, j, *jp);
    jp=ip;
    i*j%i;
    ++(*jp);
    printf("%d %d %d\n", i, j, *jp);
}
```

### Problem 5: 10 points

Implement the function described below.

```
void pEvenDouble(int firstD, int secondD, int* tdArray);
```

This prototype is for a function that accepts a 2-d array of integers (tdArray) with dimensions given by firstD and secondD. The function prints only those elements of the array for which all of the indices are even (for this task, 0 is defined as even). Hence, this function would print the values in [4][2] and [8][0] but not [2][1] or [7][2]. Implement this function using only one loop and without using array indices. (If you cannot remember, or do not know, something you believe you need to know for this implementation, invent something. Then carefully explain exactly what you invented.)

### Problem: 10 points

UNIX Matching:

Match unix command in the left column to description in the right column. Each correct answer is 1 point, with a max of 10 points. Specific instructions:

1. On your answer sheet write the numbers 1..11 which correspond to the unix commands.
2. To indicate a match place the letter to the left of the description next to a number on your answer sheet.
3. Some commands may not have an appropriate description. In that case write NA
4. Some descriptions may be used more than once

NUM	UNIX Command	Letter	Description
1	cp xx yy	A	append file named yy to the end of file xx
2	apropos xx yy	B	show the first 10 lines of the file xx
3	cat xx >> yy	C	allow you and your group to read xx
4	echo /*	D	change to the UNIX root directory
5	cat < xx   sort	E	show the file xx in sorted order
6	cat xx > yy	F	show any process whose name contains 'xx' that is running on this device
7	ps -aux   grep xx	G	copy the file xx into yy
8	chmod 640 xx	H	list the contents of your home directory
9	which xx yy	I	show the last 500 commands you executed
10	history   sort	J	show all commands relevant to 'xx'
11	head xx   cat	K	show the first command in PATH named 'xx'

