

CMSC 212 Midterm #1 (Spring 2005)

Name _____

Signature _____

Discussion Section Time (circle one):

12:00 1:00 2:00 3:00

Asad

Konstantin

- (1) This exam is closed book, closed notes, and closed neighbor. No calculators are permitted. Violation of any of these rules will be considered academic dishonesty.
- (2) You have 70 minutes to complete this exam. If you finish early, you may turn in your exam at the front of the room and leave. However if you finish during the last ten minutes of the exam please remain seated until the end of the exam so you don't disturb others. Failure to follow this direction will result in points being deducted from your exam.
- (3) Write all answers on the exam. If you need additional paper, we will provide it. Make sure your name is on any additional sheets.
- (4) Partial credit will be given for most questions assuming we can figure out what you were doing.
- (5) Please write neatly. Print your answers, if that will make your handwriting easier to read. If you write something, and wish to cross it out, simply put an X through it. Please clearly indicate if your answer continues onto another page.

Question	Possible	Score
1	16	
2	15	
3	15	
4	18	
5	21	
6	15	
Total	100	

1.) [16 points] Define and explain the following terms:

a) union (as a C language construct)

b) Program Counter

c) Bit shift operator

d) Function prototype

- 2.) [15 points] Give the exact output that would be produced by the following code. You do not need to worry about the exact location of any whitespace characters since none of the field width specifiers are given. You only need to worry about exactly what text appears on which lines.

```
#include <stdio.h>
#define ARR_SIZE 12

typedef struct{
    int size;
    int arr[ARR_SIZE];
}SType;

int main(void){
    char name[ARR_SIZE] = "Jeff Jones";
    SType s1 = {3, {7, 2, 4}};
    int *ipntr;
    char *cpntr;
    SType *spntr;

    ipntr = s1.arr;
    printf("%d and %d\n", ipntr[1], *ipntr);

    cpntr = &name[5];
    printf("%s\n", cpntr);
    printf("%c and %c\n", *(cpntr + 3), *(cpntr - 4));

    cpntr++;
    printf("%c and %c\n", name[3], cpntr[3]);

    spntr = &s1;
    printf("%d and %d\n", spntr->size, *(spntr->arr + 2));

    return 0;
}
```

3.) [15 points] Project 1

a) In project 1B you needed translate a label use into the memory address where it was defined. How did you handle this (be specific about how you handled labels that were used before being defined and defined before being used)? If you didn't get this working in your project, explain how planned to do it.

b) You have been assigned to extend the computer from project #1 with a new instruction that adds a constant (between 0 and 65,535) to the value stored in any one of the 16 registers and then puts the result in any of the registers 2-15. Explain how you would add this instruction to the machine. Make sure to indicate if a new op code is needed or could you extend an existing one like the two versions of Load (if you need a new op-code, recall 9 is unused). Also, explain how the fields of the instruction would be used.

4.) [18 points] Use the structure on the last page for the definitions of the types used here - that page can be torn off, but make sure you write your name on it and submit it with your exam paper. For each of the following questions, the first line gives the declaration of a variable, and the second is an expression using that variable.

i) You need to fill in the blank first to say if that expression would be valid or not

ii) If the expression does use the variable name declared in a legal way, you must then also tell the type the expression is.

a) `PersonTy p;`
`p -> name[0]`

b) `HourlyTy h;`
`h.numofHrs`

c) `CompanyTy x;`
`*(x.p -> name)`

d) `PArr p;`
`*p.age`

e) `WageTy w;`
`w.numofHrs[1]`

f) `PersonTy *x;`
`x.pay.weeklysalary`

5.) [21 points] Use the types defined on the last page of this exam to write each of the following functions. The last page of the exam can be torn off the exam, but make sure you put your name on that paper and submit it with your exam when you are finished. You must assume the prototype given is already present (in the .h file), and you must give the complete implementation that would appear in the corresponding .c file.

- a) A function that calculates and returns the pay for that one week of the one person passed as the argument.

```
float CalcPay(PersonTy);
```

- b) A function that prints firstname lastname. You may assume the name is stored (in a good null terminated string) as lastname, firstname. You may also assume there is exactly one comma stored (it should not be printed).

```
void printname(StrTy);
```

- c) A function that gives a 3% raise for all company employees who are salaried employees and a 5% raise for all hourly employees.

```
void giveRaises(CompanyTy *);
```

6.) [15 Points] UNIX and Make

- a) Given the partial files listed below, write a Makefile to build the program application from the files `file1.c`, `file2.c`, and `main.c`.

```
stuff.h:                                file1.c:
    #include "helper.h"                  #include "stuff.h"
    ..                                    ..

main.c:                                  file2.c:
    #include "stuff.h"                   #include "helper.h"
    ..                                    ..
```

- b) If you were to run the following sequence of UNIX commands, what is in the file `dir1/stuff`?

```
cd
mkdir dir1
cp myFile stuff
cd dir1
cp ../otherStuff stuff
```

- c) What is the difference between `<` and `>` when used on a UNIX command line?

Name: _____

The types defined on this page will be used for two of the questions on this exam. This page can be torn off the exam paper to make it easier for you to see the definitions while you answer the questions. Make sure you write your name on this paper, put it inside your exam, and submit it with your exam paper when you are finished.

```
#define ARRSIZE 20

typedef char StrTy[ARRSIZE];

typedef struct{
    int numofHrs[7]; /*number of hours worked each day of the 7 in a week*/
    float perhourrate; /* how much the person gets paid per hour */
} HourlyTy;

typedef union{
    float    weeklysalary; /*weekly salary of a salaried worker*/
    HourlyTy  hrlysalary; /*pay for hourly worker as defined above*/
} WageTy;

typedef struct {
    StrTy name;          /*person's name */
    int    age;          /* age of person in years */
    int    emptytype;   /* 0 for employee on weekly salary or
                        non-zero for hourly employee*/
    WageTy pay;         /* pay as defined above */
} PersonTy;

typedef PersonTy PArr[ARRSIZE];

typedef struct{
    StrTy name;          /* the name of the company the employee list is for */
    PArr  p;            /* the array of employees for this company */
    int  count;         /* the number of employees in this company */
} CompanyTy;
```

This page intentionally blank.