

The Midterm contains 4 pages and 8 questions. Total points possible are 100.

Submit the Midterm to the canvas assignment marked Midterm. The solutions/programs should be in a folder marked midterm and the contents zipped. Use Netbeans, delete the build and dist folders before zipping. Do the best you can and turn in as much as you can.

I just want one program for the solution to the problems that follow. The program should prompt the user for which problem solution to display. Use a do-while and switch construct like the included menu program.

Develop each in a separate project. Then combine into one project when done. You are to include each project as well as the combined menu project when submitting your midterm.

1. (15 points) Develop an application using a structure for a customer that will determine if he/she has exceeded his/her checking account balance. For each customer, the following facts are available and the structure should include in the following order:
 - a. Account number (Five digits only, i.e. test for this)
 - b. Name
 - c. Address
 - d. Balance at the beginning of the month
 - e. Record all checks written by this customer this month Loop until customer is through entering checks.
 - f. Record all deposits credited to this customer's account this month. Loop until customer is through entering deposits.

Dynamically create the structure.

The program should input each of these facts from input dialogs, store in a structure, calculate the new balance, display the new balance and debit the account \$25 if overdrawn. Also, if overdrawn, tell the customer the additional \$25 fee has been accessed and what the balance would be with this fee included. Make sure to output the contents of the structure.

2. (15 points) Develop an application using an array of structures created dynamically that will determine the gross pay for any number of employees input. i.e. ask up front how many employees you are inputting. The company pays "straight-time" for the first 40 hours worked, double time for all hours worked in excess of 40 hours but less than 60 hours, and triple time for any hours worked over 60 hours.

The program should be able to loop and input the employee's name, hours worked, and the rate of pay. Once this has been done then output this information including the

gross pay in the form of a paycheck. The process starts all over again until you input an invalid rate of pay or hours worked. This means a negative rate of pay or negative number of hours worked is not acceptable.

Print it out like an actual paycheck with English language representing the check amount. Printout like a company Paycheck with format:

- Company
- Address
- Name: Amount: numerical
- Amount: English of numerical value
- Signature Line:

Hint: Utilize the Roman Numeral Conversion Program to help with producing the check.

3. (15 points) I have written a program for you to complete. This problem is not to be included with the menu. Simply print when I run the menu program that it is included in another project. You are to write a program that outputs the mean, median, and mode. Your task is to complete the function. This will use structures to pass information from one function to the print function.

```
Stats *stat(const Array *array){
    //Non-working stub to be completed by the student
    cout<<endl<<"stat function to be completed by the student"<<endl;
    Stats *stats=new Stats;
    stats->mode=new Array;
    stats->mode->size=0;
    int nModes=0;
    stats->mode->data=new int[nModes];
    stats->modFreq=0;
    stats->median=0;
    return stats;
}
```

I have also supplied the expected outputs you can compare against in a StatResults.txt file. Note: The only part of the program you are to change is the stat function. Nothing else is to be touched!

4. (10 points) A company wants to transmit data over the telephone, but they are concerned that their phones may be tapped. All of their data is transmitted as four-digit integers. However, they utilize the digits 0,1,2,3,4,5,6,7 only. They have asked you to write a program that will encrypt their data so that it may be transmitted more securely. Your application should read a four-digit integer with the numbers 0 to 7 entered by the user in an input dialog and encrypt it as follows: Replace each digit by (the sum of that digit plus 3 modulus 8). Then swap the first digit with the third, and swap

the second digit with the fourth. Then print the encrypted integer. Write a separate application that inputs an encrypted four-digit integer and decrypts. If you encrypt a number, you should be able to get the same number decrypted back when the user does this. Note: sometimes errors occur and the digits 8, and 9 might appear. Detect this condition and flag the error. Can't read in numbers individually. All the digits must be read with one cin statement.

5. (10 points) Exploration of datatypes!

- a) Using a byte variable, what is the largest n where $n!$ =factorial that can be calculated. A factorial is simply
$$n! = 1 * 2 * 3 * 4 \dots * (n-2) * (n-1) * n$$
For instance, $1! = 1$, $2! = 2$, $3! = 6$, $4! = 24$, $5! = 120$ etc....Signed as well as unsigned.
- b) What would be the largest n for $n!$ using short, int, long, long long, float, double, etc... Use all the primitive data types that you are familiar with signed as well as unsigned.

Note: I want you to investigate this and the program you turn in should just print "cout" the final results. You don't know how to capture errors yet so run each till it overflows then back off by 1. Don't need the program that tests for this.

6. (15 points) Conversion to NASA 4 byte float format

- a) Convert the following 3 numbers to binary, octal and hex 49.1875, 3.07421875, 0.2. When done, convert the following to a float representation by the definition in class. In other words, I want an 8 digit hex number representation using the 4 byte float specification defined in class. An example .pdf is attached for you to review.
- b) Do the same for a) given they are negative values.
- c) Convert the float representations of the following into the decimal number given the definition in class. 69999902, 69999903, 966667FF

Note: This doesn't require a program, however, I want you to write a simple function that outputs the answers you did by hand with "cout". Also, scan in your calculations done by hand.

7. (10 points) Factor an input integer into it's prime numbers. Create a structure to hold the array of prime numbers. Any number between [2,65000] will be a valid number to check for prime factors.

```
struct Prime{
    unsigned short prime;
    unsigned char power;
};
struct Primes{
    unsigned char nPrimes;
```

```
        Prime *prime;  
    };
```

Output the number and it's primes i.e. $120 = 2^3 * 3^1 * 5^1$

So create a function

Primes *factor(int); Input an integer, return all prime factors

void prntPrm(Primes *); Output all prime factors

8. (10 points) Menu