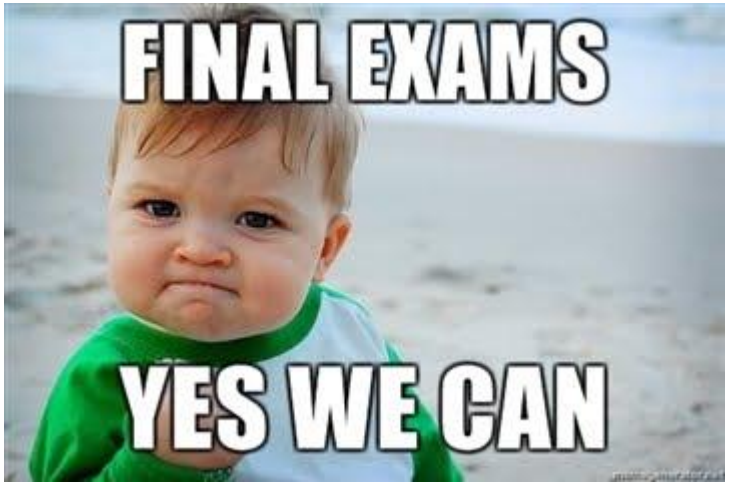
GNG1106

Chao Sun(csun014@uottawa.ca)

Contents



2013 Final Exam

2013 Final Exam

Question 1 *Suppose that variable* `x` *is declared as* `int` *type. Which one of the following statements will read the value of* x *from the keyboard?*

(A)
```
scanf("%d", x);
```

(B)
```
scanf("%d", &x);
```

(C)
```
printf("%d", x);
```

(D)
```
printf("%d", &x);
```

Answer: B

- standard I/O functions: scanf() and printf()
- conversion specifier

2013 Final Exam(Cont.)

Question 2 What will be the value of variable i after the following code segment is executed?

```c
#include<stdio.h>

int main()
{
    int x=0;
    int i;
    for(i=0;i<100;i++)
    {
        x=x+i;
        if(x>20)
            break;
    }
    printf("i = %d and x = %d", i, x);
    return 0;
}
```

2013 Final Exam(Cont.)

Answer: i = 6

- for loop
- if /else statement
- break

```c
#include<stdio.h>

int main()
{
    int x=0;
    int i;
    for(i=0;i<100;i++)
    {
        x=x+i;
        if(x>20)
            break;
    }
    printf("i = %d and x = %d", i, x);
    return 0;
}
```

i	x
0	0
1	1
2	3
3	6
4	10
5	15
6	**21**

2013 Final Exam(Cont.)

Question 3 What will the code print out?

```c
#include <stdio.h>

int foo(int a, int b)
{
    int c = 0;
    c = c+b-a;
    a=b;
    return c;
}

int main(void)
{
    int a = 1;
    int b = 2;
    int c = 3;
    c = foo(a, b);
    printf("a = %d, c = %d\n", a, c);
    return 0;
}
```

2013 Final Exam(Cont.)

- passing arguments by values
- variable scope
- calling a function
- the return value

```c
#include <stdio.h>

int foo(int a, int b)
{
    int c = 0;
    c = c+b-a;
    a=b;
    return c;
}

int main(void)
{
    int a = 1;
    int b = 2;
    int c = 3;
    c = foo(a, b);
    printf("a = %d, c = %d\n", a, c);
    return 0;
}
```

```
a = 1, c = 1

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

2013 Final Exam(Cont.)

Question 4 What will the code print out?

```c
#include <stdio.h>

int foo(int *a, int b)
{
    int c = 0;
    c = c+b-*a;
    *a=b;
    return c;
}

int main(void)
{
    int a = 1;
    int b = 2;
    int c = 3;
    c = foo(&a, b);
    printf("a = %d, c = %d\n", a, c);
    return 0;
}
```

2013 Final Exam(Cont.)

- passing arguments by reference

```c
#include <stdio.h>

int foo(int *a, int b)
{
    int c = 0;
    c = c+b-*a;
    *a=b;
    return c;
}

int main(void)
{
    int a = 1;
    int b = 2;
    int c = 3;
    c = foo(&a, b);
    printf("a = %d, c = %d\n", a, c);
    return 0;
}
```

```
a = 2, c = 1

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

2013 Final Exam(Cont.)

Question 5 What will the code print out?

```c
#include <stdio.h>

int foo(int a, int b)
{
    static int c = 0;
    c = c+b-a;
    a=b;
    printf("c in the foo() = %d\n", c);
    return c;
}

int main(void)
{
    int a = 1;
    int b = 2;
    int c = 3;
    c = foo(a, b);
    c = foo(a, b);
    c = foo(a, b);
    printf("c = %d\n", c);
    return 0;
}
```

2013 Final Exam(Cont.)

- static variable scope

```
c in the foo() = 1
c in the foo() = 2
c in the foo() = 3
c = 3

Process returned 0 (0x0)   execution time : 0.028 s
Press any key to continue.
```

```c
#include <stdio.h>

int foo(int a, int b)
{
    static int c = 0;
    c = c+b-a;
    a=b;
    printf("c in the foo() = %d\n", c);
    return c;
}

int main(void)
{
    int a = 1;
    int b = 2;
    int c = 3;
    c = foo(a, b);
    c = foo(a, b);
    c = foo(a, b);
    printf("c = %d\n", c);
    return 0;
}
```

2013 Final Exam(Cont.)

Question 6 *An array* `a` *is declared by*

```
int a[10];
```

In the expressions below, which ones are equivalent to the expression `a[5]`*?*

(A) `*(a+5)`

(B) `&(a+5)`

(C) `&(*a[5])`

(D) `*a+5`

(E) `&(*a+5)`

Answer: A

2013 Final Exam(Cont.)

Question 7 *In order to use function* `malloc`, *which one of the following lines should be contained in your code?*

(A) `#include <stdio.h>`

(B) `#include <stdlib.h>`

(C) `#include <string.h>`

(D) None of above

Answer: B

2013 Final Exam(Cont.)

Question 8 *The C code of a program is given below. How can you run this program so that what the program does is precisely printing to the screen the string "*`I LIKE EXAMS`*"?*

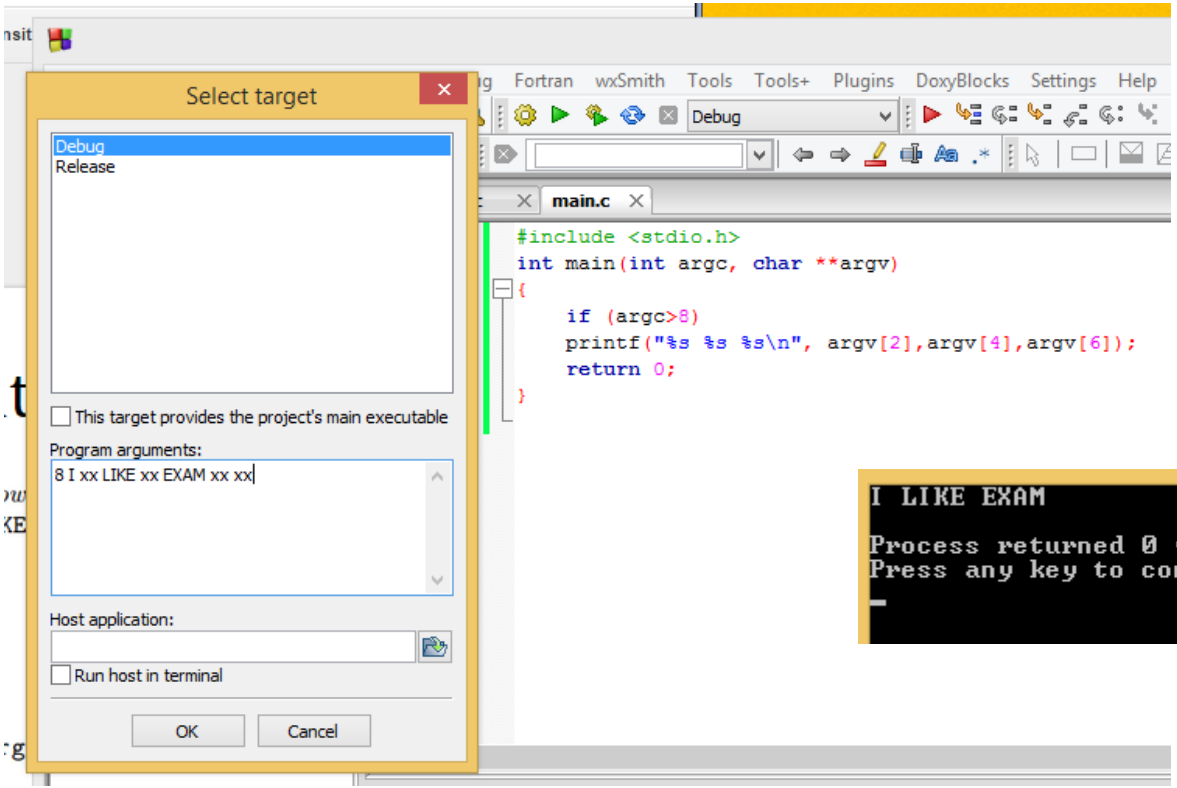
```
#include <stdio.h>

int main(int argc, char **argv)
{
        if (argc>8)
                printf("%s %s %s\n", argv[2], argv[4], argv[6]);
        return 0;
}
```

2013 Final Exam(Cont.)

- With argc (argument count) and argv (argument vector) you can get the number and the values of passed arguments when your application has been launched.

- The first argument is the number of parameters passed plus one to include the name of the program that was executed to get those process running. Thus, argc is always greater than zero and argv[0] is the name of the executable (including the path) that was run to begin this process

2013 Final Exam(Cont.)




```
I LIKE EXAM

Process returned 0 (0x0)   execution time : 1.049 s
Press any key to continue.
```

2013 Final Exam(Cont.)

Question 9 *What is the execution result of the following program?*

```
#include <stdio.h>

int foo(int *a, int b)
{
        static int c=0;
        c=c+b-*a;
        *a=b;
        printf("in foo: *a=%d, b=%d, c=%d\n", *a, b, c);
        return c;
}

int main(void)
{
        int a=1;
        int b=2;
        int c=3;
        c=foo(&a, b);
        printf("in main: a=%d, b=%d, c=%d\n", a, b, c);
        c=foo(&c, b);
        printf("in main: a=%d, b=%d, c=%d\n", a, b, c);
        return 0;
}
```

```
In foo: *a =2, b=2, c=1
In main: a =2, b=2, c=1
In foo: *a =2, b=2, c=2
In main: a =2, b=2, c=2

Process returned 0 (0x0)
Press any key to continue.
```

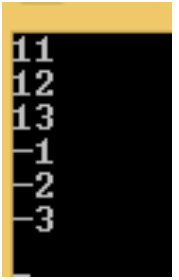
- passing argument by reference
- static variables

2013 Final Exam(Cont.)

```c
#include<stdio.h>
void func1(int *b)
{
    int *a;
    a = (int *)malloc(3*sizeof(int));
    a[0] = 1;
    a[1] = 2;
    a[2] = 3;
    b=a;
}
void func2(int **b)
{
    int *a;
    a = (int *)malloc(3*sizeof(int));
    a[0] = -1;
    a[1] = -2;
    a[2] = -3;
    *b=a;
}
int main()
{
    int *p;
    int i;
    p = (int *)malloc(3*sizeof(int));
    p[0] = 11;
    p[1] = 12;
    p[2] = 13;
    func1(p);
    for (i=0;i<3;i++)
        printf("%d\n", p[i]);
    func2(&p);
    for (i=0;i<3;i++)
        printf("%d\n", p[i]);
    return 0;
}
```

2013 Final Exam(Cont.)

```c
#include<stdio.h>
void func1(int *b)
{
    int *a;
    a = (int *)malloc(3*sizeof(int));
    a[0] = 1;
    a[1] = 2;
    a[2] = 3;
    b=a;
}
void func2(int **b)
{
    int *a;
    a = (int *)malloc(3*sizeof(int));
    a[0] = -1;
    a[1] = -2;
    a[2] = -3;
    *b=a;
}
int main()
{
    int *p;
    int i;
    p = (int *)malloc(3*sizeof(int));
    p[0] = 11;
    p[1] = 12;
    p[2] = 13;
    func1(p);
    for (i=0;i<3;i++)
        printf("%d\n", p[i]);
    func2(&p);
    for (i=0;i<3;i++)
        printf("%d\n", p[i]);
    return 0;
}
```



2013 Final Exam(Cont.)

Question 11 (9 points) *You are to write a function with prototype* `float func(void)`. *The function, when called, will keep prompting the user to enter* **positive** *float numbers until the user enters a negative number. The function discards the largest number and the smallest positive numbers that the user has entered and returns the average of the remaining positive numbers. (***You may get an additional 4 bonus points if your function uses the minimal amount of memory and takes the minimal amount of time.***)**

2013 Final Exam(Cont.)

```
float func(void)
{
    float  new_num = 0.0, count = 0.0, sum = 0.0, avg = 0.0, max = 0.0, min = 100000.0;
    printf("Please enter a positive float number, enter a negative number to exit\n");
    scanf("%f", &new_num);
    while (new_num > 0)
    {
        if(new_num > max)
            max = new_num;
        if(new_num < min)
            min = new_num;
        sum = sum + new_num;
        count++;
        printf("Please enter a positive float number, enter a negative number to exit\n");
        scanf("%f", &new_num);
    }
    avg = (sum - max - min)/(count-2);
    printf("Negative number entered, exit\n");
    printf("The maximum and minimum numbers entered are %f and %f\n", max,min);
    printf("The average of the rest positive numbers is %f\n",avg);
    return avg;
}
```

2013 Final Exam(Cont.)

Question 12 (9 points) *A text file contains 1000 float-type numbers in the range from -1000 to 1000. Write a program that checks how many numbers in the file that is in the range from -1 to 1.*

2013 Final Exam(Cont.)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    FILE *src = NULL;
    float num;
    int counter = 0;
    src = fopen("num.txt", "r");

    if(src == NULL)
    {
        printf("Error\n");
        exit(0);
    }
    else
    {
        while(!feof(src))
        {
            fscanf(src, "%f", &num);
            if(num<1 && num >-1)
            {
                counter++;
            }
    }
    printf("There are %d numbers between -1 to 1", counter);
    fclose(src);
    return 0;
    }
}
```

2012 Final Exam

2012 Final Exam

Question 1 *(5×13=65 points) Answer the following questions. Justify your answer.*

1. What will the following code print?

```
#include <stdio.h>

int main(void)
{
        if (0)
                printf("Good Bye World,");
        else
                printf("Hello World,");

        printf(" if the world ends on December 21.\n");

        return 0;
}
```

2012 Final Exam(Cont.)

2. When using a `char` array to store string “Hello”, the array length need to be 5 or higher. Is this statement correct?

2012 Final Exam(Cont.)

3. What is the value of variable a after the following code segment is executed?

```
double a;
a=3/5;
```

Answer: a = 0.000000

2012 Final Exam(Cont.)

4. After the following code segment is executed, regarding the value of variable `c`, which one of the following is correct?

   (a) 0;

   (b) a positive random number;

   (c) a negative random number;

   (d) RAND_MAX

```
int a1, b1, a2, b2, c;
srand(123);
a1=rand();
b1=rand();
srand(123);
a2=rand();
b2=rand();
c=a1+b1-a2-b2;
```

Answer: (a)

Same seeds generate same sequences of random numbers. Please check the values of a1,b1,a2,b2 .

2012 Final Exam(Cont.)

5. What will be printed when the following code is executed?

```
#include <stdio.h>

void swap (int a, int b)
{
        int tmp;
        tmp=a;
        a=b;
        b=tmp;
}

int main(void)
{
        int a=10;
        int b=20;
        swap(a, b);
        printf("a=%d, b=%d\n", a, b):
        return 0;
}
```

2012 Final Exam(Cont.)

Answer: a = 10, b=20

5. What will be printed when the following code is executed?

```
#include <stdio.h>

void swap (int a, int b)
{
        int tmp;
        tmp=a;
        a=b;
        b=tmp;
}

int main(void)
{
        int a=10;
        int b=20;
        swap(a, b);
        printf("a=%d, b=%d\n", a, b):
        return 0;
}
```

2012 Final Exam(Cont.)

6. Suppose that in the code above, the line "swap(a, b)" in the main function is replaced with "*(&(a)) =b", then what do you expect? (Will the code compile? If it does not, identify the error; if it does compile, what is the result of the execution?)

5. What will be printed when the following code is executed?

```
#include <stdio.h>

void swap (int a, int b)
{
        int tmp;
        tmp=a;
        a=b;
        b=tmp;
}

int main(void)
{
        int a=10;
        int b=20;
        swap(a, b);
        printf("a=%d, b=%d\n", a, b):
        return 0;
}
```

2012 Final Exam(Cont.)

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{

    int a=10;
    int b=20;
    *(&a) = b;
    printf("a = %d and b = %d", a,b);
    return 0;
// The answer is a = 20, b=20
}
```

Answer: a = 20, b=20

2012 Final Exam(Cont.)

7. The function `getNext()` is given as follows.

```
int getNext(void)
{
        static int prev=1;
        static int current=0;
        int tmp;
        tmp=prev+current;
        prev=current;
        current=tmp;
        return current;
}
```

When the function is called for the 6-th time, what is its return value?

2012 Final Exam(Cont.)

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

int getNext()
{
    static int prev = 1;
    static int current = 0;
    int tmp;
    tmp = prev+current;
    prev = current;
    current = tmp;
    return current;
}
int main()
{

    int i; int result;
    for(i=0;i<6;i++)
    {
        result = getNext();
    }
    printf("The result is %d\n", result);
    return 0;
// The answer is 8
}
```

2012 Final Exam(Cont.)

8. Suppose that you are to read a binary file with name **myBinFile.bin**, when opening the file, you use the following line of c code (where variable **fp** has been declared as a FILE pointer):

```
fp=fopen("myBinFile.bin", ?);
```

The question mark in the line above is meant to be a string. What is the string that you should use?

Answer: "rb" or "rb+"

mode (text/binary)	Description
"r"/"rb"	Open a file for reading. The file must exist.
"w/wb"	Create an empty file for writing. If a file with the same name already exists its content is erased and the file is considered as a new empty file.
"a"/"ab"	Append to a file. Writing operations append data at the end of the file. The file is created if it does not exist.
"r+"/"rb+"	Open a file for update both reading and writing. The file must exist.
"w+"/"wb+"	Create an empty file for both reading and writing.
"a+"/"ab+"	Open a file for reading and appending.

2012 Final Exam(Cont.)

9. What will be printed by the following code?

```
#include <stdio.h>

int main(void)
{
        int a[]={1, 2, 3, 4, 5};
        printf("%d\n", a[3]);
        return 0;
}
```

Answer: 4

2012 Final Exam(Cont.)

10. What will be printed by the following code?

```
#include <stdio.h>

int main(void)
{
        int a[]={1, 2, 3, 4, 5};
        int b[]={2, 4, 6, 8, 10};
        int i;
        *a=*b;
        for (i=0; i<3;i++)
        {
                printf("a=%d, b=%d\n", a[i], b[i]);
        }
        return 0;
}
```

Answer:

a =2 , b= 2

a = 2, b= 4

a = 3, b= 6

2012 Final Exam(Cont.)

11. What will be printed by the following code?

```
#include <stdio.h>

int main(void)
{
        int a[]={1, 2, 3, 4, 5};
        int *b;
        int i;
        b=&a[2];
        for (i=0; i<3;i++)
        {
                printf("b=%d\n", b[i]);
        }
        return 0;
}
```

Answer:

b= 3

b= 4

b= 5

2012 Final Exam(Cont.)

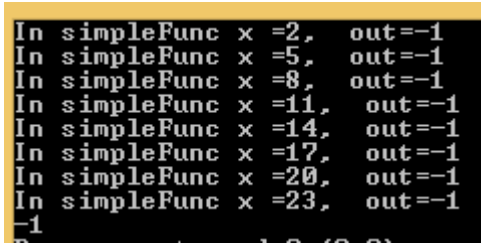
12. The function **simpleFunc** is given as follows.

```
int simpleFunc(int x)
{
        int out=-1;
        if (x>=3)
                out=simpleFunc(x-3);
        if (x<0)
                out=-1;

        return out;
}
```

What is the return value of the function if the function is called as `simpleFunc(23)`?

2012 Final Exam(Cont.)

```c
#include<stdio.h>
int simpleFunc(int x)
{
    int out = -1;
    if(x>=3)
        out = simpleFunc(x-3);
    if(x<0)
        out = -1;
    printf("In simpleFunc x =%d,  out=%d\n", x, out);
    return out;
}
int main()
{
    int i = simpleFunc(23);
    printf("%d", i);
    return 0;
}
```



2012 Final Exam(Cont.)

13. Determine if the following code segment has syntax error?

```
char *buff;
buff=(char *)malloc(sizeof(int));
*((int *)buff)=10;
```

Answer: No error.

2012 Final Exam(Cont.)

Question 2 *(10 points) Write the c code for the following program.*

The program keeps prompting the user to enter (from the keyboard) positive integers until the user enters "0". The program then prints the average value of all entered positive integers. Note that the average of a set of integers are typically not an integer. (There is no need to check if the user enters a valid input to the program, namely, assuming that the user always uses the program properly.)

2012 Final Exam(Cont.)

```c
#include <stdio.h>
int main ()
{
    int i;
    float ave;
    int counter=0, sum=0;
    printf("Please enter a series of positive integer numbers.\n");
    printf("Enter '0' to end.\n");
    while(1)
    {
        scanf("%d",&i);
        if(i==0)
            break;
        else
        {
            counter++;
            sum+=i;
        }
    }
    if(counter == 0)
    {
        printf("No number entered.\n");
    }
    else
    {
        ave=(float)sum/counter;
        printf("The average value is %f\n",ave);
    }

    return 0 ;
}
```

2012 Final Exam(Cont.)

Question 3 *(10 points) Write the c code for the following function.*

The function is to compare two integer vectors of the same length and returns the number of places at which the two vectors differ. For example, vector $(0, 2, 2, 3, 5)$ and vector $(1, 2, 3, 3, 5)$ differ in two places. Apparently, the vectors should be supplied to the function as input. Note that the length of the vectors may be arbitrary. Implement the function in any way you wish as long as it does the job.

2012 Final Exam(Cont.)

```
int getDifference(int *a, int *b, int length)
{
    int i;
    int out=0;
    for (i=0;i<length; i++)
    {
        if (a[i]!=b[i])
            out=out+1;
    }
    return out;
}
```

2012 Final Exam(Cont.)

Question 4 *(7 points) Write the c code for the following function.*

The function has the prototype

`double **matrixMultiply(double **A, int nRowsInA, int nColsInA, double **B, int nRowsInB, int nColsInB)`

The function performs matrix multiplication, namely it multiplies (not necessarily square) matrices A and B and return matrix AB. Matrices need to be represented as pointers to pointers to double, as seen from the function prototype. The arguments `nRowsInA, nColsInA, nRowsInB` and `nColsInB` refer to respectively the number of rows of A, the number of columns of A, the number of rows of B and the number of columns of B.

2012 Final Exam(Cont.)

```c
#include <stdlib.h>

double **matrixMultiply(double **A, int nRowsInA, int nColsInA, double **B, int nRowsInB,
int nColsInB)
{
    int i, j, k;
    double **out;
    out=(double **)malloc(nRowsInA*sizeof(double *));
    for (i=0;i<nRowsInA;i++)
        out[i]=(double *)malloc(nColsInB*sizeof(double));

    for (i=0;i<nRowsInA; i++)
        for (j=0;j<nColsInB; j++)
            out[i][j]=0;


    for (i=0;i<nRowsInA; i++)
        for (j=0;j<nColsInB; j++)
            for (k=0;k<nColsInA;k++)
                out[i][j]=out[i][j]+A[i][k]*B[k][j];

    return out;
}
```

2012 Final Exam(Cont.)

Question 5 *(15 points) Complete the following programming task.*

A text file with name `students.dat` contains the following content.

```
10
james       bond        1234789
jacky       chan        9088712
jet         li          2119987
an          lee         9383783
james       smith       9838377
David       Webb        9988123
jason       borne       3467222
yong        chang       9383866
will        smith       2324242
jessy       Zhang       8498944
```

Note that except for the first line, every line in the file contains the record of a student (in the order of first name, last name and student number). The first line contains an integer that indicates how many records are contained in the file. (Verify that there are 10 student records in the above list).

You are to complete a program that reads the `students.dat` file (and other such files), and creates another text file that is in the same format except that

1. All letters are in capitalized.
2. the records are sorted in ascending order of the student numbers.

The program is partially written and the source code is given below. The `main` function of the program is not to be changed, and all you need to do is to fill in the details for all other functions.

```c
struct student *readStudentRecordFile(char * fileName, int * numOfStudents)
{
    int i;
    struct student * studentList = NULL;
    FILE * fpr = NULL;
    char firstName[MAXNAMELEN];
    char lastName[MAXNAMELEN];
    if (!fileName || !numOfStudents) {
        return NULL;
    }
    fpr = fopen(fileName,"r+");
    if (!fpr) {
        fprintf(stderr, "Cannot access student record file %s.\n", fileName);
        return NULL;
    }
    fscanf(fpr,"%d", numOfStudents);
    if (*numOfStudents == 0) {
        fprintf(stderr, "Student record file %s contains 0 student record.\n", fileName);
        fclose(fpr);
        return NULL;
    }
    studentList = (struct student *)calloc((*numOfStudents), sizeof(struct student));
    if (!studentList) {
        fprintf(stderr, "No memory.\n");
        fclose(fpr);
        return NULL;
    }
    for(i = 0;i < *numOfStudents; i++){
        fscanf(fpr,"%s %s %d", firstName, lastName, &(studentList[i].studentNumber));
        firstName[MAXNAMELEN - 1] = '\0';
        lastName[MAXNAMELEN - 1] = '\0';
        studentList[i].firstName = strdup(firstName);
        studentList[i].lastName =strdup(lastName);
    }
    fclose(fpr);
    return studentList;
}
```

```c
void writeStudentRecordFile(char*fileName , struct student * allStudents, int numOfStudents)
{
    int i;
    FILE *fpw = NULL;

    fpw = fopen(fileName,"w+");
    if (!fpw) {
        fprintf(stderr, "Cannot create student record file %s.\n", fileName);
        return;
    }
    fprintf(fpw,"%d\n",numOfStudents);

    for(i = 0;i < numOfStudents;i++){
        if (allStudents[i].firstName && allStudents[i].lastName) {
            fprintf(fpw,"%s\t%s\t%d\n",allStudents[i].firstName, allStudents[i].lastName, allStudents[i].studentNumber);
        }
    }
    fclose(fpw);
    fprintf(stdout, "Done processing the student records. Result is stored in %s\n", fileName);
}
```

```c
struct student *sortStudents(struct student * allStudents, int numOfStudents)
{
    int tmpNumber;
    char * tmpFirstName;
    char * tmpLastName;

    int i,j;
    for(i=0;i<numOfStudents;i++){
        for(j=i+1;j<numOfStudents;j++){
            if(allStudents[i].studentNumber>allStudents[j].studentNumber){
                tmpNumber = allStudents[i].studentNumber;
                tmpFirstName = strdup(allStudents[i].firstName);
                tmpLastName = strdup(allStudents[i].lastName);

                allStudents[i].studentNumber = allStudents[j].studentNumber;
                free(allStudents[i].firstName);
                free(allStudents[i].lastName);
                allStudents[i].firstName = strdup(allStudents[j].firstName);
                allStudents[i].lastName = strdup(allStudents[j].lastName);

                allStudents[j].studentNumber = tmpNumber;
                free(allStudents[j].firstName);
                free(allStudents[j].lastName);
                allStudents[j].firstName = tmpFirstName;
                allStudents[j].lastName = tmpLastName;
            }
        }
    }

    return allStudents;
}
```

```
void capitalizeNames(struct student*allStudents, int numOfStudents)
{
    int i,j;
    int len;
    for(i=0;i<numOfStudents;i++){

        len=strlen(allStudents[i].firstName);
        for(j=0;j<len;j++){
            allStudents[i].firstName[j] = (char) toupper((int) allStudents[i].firstName[j]);
        }

        len=strlen(allStudents[i].lastName);
        for(j=0;j<len;j++){
            allStudents[i].lastName[j] = (char) toupper((int) allStudents[i].lastName[j]);
        }
    }
}
```

Do not forget to free the memory!

Thank you for your support during this semester!