BM 593 Numerical Methods & C Programming

1st week Basic Syntax of C Language Simplest examples of a C code

```
/* Anything in between these slash-asterisk couple is treated as Comment */
#include <stdio.h>
void main(){
     printf("Hello World\n");
}
```

```
#include <stdio.h>
#define OutputMessage "Hello World\n"
void main(){
     printf(OutputMessage);
}
```

Declaration and Definition of Variables

```
int x=1;
float num;
int i,j,k;
double mean, variance;
char c;
unsigned m; /* or unsigned int : 4 bytes  */
unsigned char; /* 1 byte */
unsigned long; /* 8 byte */
unsigned short; /* 2 byte */

/* Array Definitions */
char string[80];
int image[256][256];
double volume_image[64][64][64];

struct student
{   int no;
    char name[30];
    char address[30];
    int year;
}
```

```
struct student Ali, Can;
struct student class[20];

Ali.name = "Ali Ozay";
Ali.address = "BU BME";
Ali.year = 2001;

typedef struct student Student;
Student BM593Class[15];
BM593Class[0].name="Murat";
```

Pointers

```
int *p /* p is an address of an integer location */
*p = 1; /* Assignment */
p+1
```

denotes the address of the next integer location.

```
int x;
p=x /* p points to x */
x=*p /* copies the value of p to x */

int z[10];
p=&z[0]; /* p points to z[0] */
```

Relations between Pointers and Arrays

```
int a[10];
int *p;
p=&a[0];  or p=a;  /* p points the beginning address of a */
x=*p; /* the content of a[0] is copied into x */
x=*(p+1) /* the content of a[1] is copied into x */
x=a[i]; /* same as  x=*(a+1); */

p=a;  /* a=p is not allowed */
p++;  /* a++ is not allowed */
```

Statements: for

```
 for (i=0; i<10; i++)
   x[i]=3;

 for (i=0; i<10; i++){
    for (j=0; j<10; j++)
       a[i][j]=i+2*j;
    x[i]=i;
 }


 for (i=0,k=0; i<10;i++,k++)
    x[i]=i;
```

Statements: if

```
 if (i==3)
   x[i]+=x[i];


 if (i<3)
   x[i]+=3.;
 else{
   x[i]-=5.;
   i--;
 }
```

Statements: while

```
 while (i > 0) {
   x[i]=i*2.;
   i++;
 }
```

Statements: do while

```
 do {
   x[i]=i;
   i--;
 }while (i>4;)
```

Statements: switch

```
switch (i) {
 case 'q' : printf ("It is quitting \n");
  break;
 case 'r' : printf ("It is reading \n");
  break;
 break : printf ("It is doing nothing \n");
}
```

Statements

break

```
  for (i=0; i<5; i++)
    if (x[i]>3)
      break;
```

Statements

continue

```
for (i=0,sum=0; i<5; i++){
    if (i>3) continue;
    sum+=i;
  }
```

Dynamic Memory Allocation

vector allocation

```
#include <stdlib.h>
void main(){

  int *int_vector;
  int_vector = (int *) malloc(10*sizeof(int));  /* or */
  /* int_vector = (int *) calloc(10,sizeof(int)); */
  /* to free the dynamic memory */
  free (int_vector);
}
```

Dynamic Memory Allocation

matrix allocation

```
#include <stdlib.h>
void main(){

  int **int_matrix,i;
  int_matrix = (int **) malloc(10*sizeof(int*));  /* or */
  /* int_matrix = (int **) calloc(10,sizeof(int*)); */
  for (i=0;i<10; i++)
    int_matrix[i] = (int *) malloc(20*sizeof(int)); /* or */
  /* for (i=0;i<10; i++)
        int_matrix[i] = (int *) calloc(20*sizeof(int)); */

  /* to free the dynamic memory */
  for (i=0;i<10; i++)
    free (int_matrix[i]);
  free(int_matrix);
}
```

Functions

```
#include <stdio.h>
int power (int , int )

void main(){
  int i;
  for (i=0; i<10; i++)
  printf ("%d %d\n", i, power(2,i));
}

int power (int m, int n)
{
  int i,p;
  p=1;
  for (i=1; i<=n; i++)
    p*=m;
  return p;
}

#include <stdio.h>
void power (int , int , int *)

void main(){
  int i,p;
  for (i=0; i<10; i++){
    power(2,i,&p);
    printf ("%d %d\n", i,p);
  }
}

void  power (int m, int n, int *p)
{
  int i;
  *p=1;
  for (i=1; i<=n; i++)
    *p*=m;
}

#include <stdlib.h>

void main(){

  int i,**int_matrix;

  int_matrix = (int **) calloc(10,sizeof(int*));
  for (i=0;i<10; i++)
    int_matrix[i] = (int *) malloc(20*sizeof(int));

  scaleImage (int_matrix, 100, 10, 20);

  /* to free the dynamic memory */
  for (i=0;i<10; i++)
    free (int_matrix[i]);
  free(int_matrix);
}

  void scaleImage (int ** int_matrix, int scale, int row, int column){
    int i,j;

    for (i=0;i<row,i++)
      for (j=0;j<column;j++)
        int_matrix[i][j]*=scale;
  }
```

Standard Input/Output

```
#include <stdio.h>

void main(){
   long unsigned int i,*p;

   printf("Enter the value of x\n");
   scanf("%lu",&x);
   printf("the value of x is %lu\n",x);
   printf("Enter the value of p\n");
   scanf("%lu",p);
   printf("the value of p is %lu\n",*p);
}
```

File Input/Output for ASCII TEXT FILES

```
#include <stdio.h>

void main(){

  FILE *f1;
  FILE *f2;
  double x[100];

  f1=fopen("c:\\data\\input.dat","r");
  f2=fopen("c:\\data\\output.dat","w");

  for (i=0;i<100;i++){
    fscanf(f1,"%lf",&x[i]); /* or fscanf(f1,"%lf",x+i) */
    fprintf(f2,"%lf",x[i]); /* or fprintf(f2,"%lf",*(x+i)) */
  }

  fclose(f1);
  fclose(f2);
}
```

File Input/Output for BINARY DATA FILES

```
#include <stdio.h>
void main(){

  FILE *f1;
  FILE *f2;
  double x[100];

  f1=fopen("c:\\data\\input.dat","r+b");
  f2=fopen("c:\\data\\output.dat","w+b");

  for (i=0;i<100;i++){
    fread(x,sizeof(double),100,f1);
    fwrite(x,sizeof(double),100,f2);
  }

  fclose(f1);
  fclose(f2);
}
```

File Operations

```
fseek(f1,100); /* puts the current file pointer 100 bytes ahead of the beginning of the file
rewind(f1); /* sets the current file pointer to the beginning of the file
when some data has already been written to or read from the file */
```