Week 3: C Programming Practical

If statements

If statements are used to make decisions. The following code identifies if a mark makes the grade:

```
#include <stdio.h>
#include <conio.h>
int main(){
int mark=50;
if (mark >= 40)
{
printf("Pass");
}
else {
printf("Fail");
}
getch();
return 0;
}
```

We can make multi-way decisions using else-if statements

```
#include <stdio.h>
#include <conio.h>
int main(){
int mark=50;
if (mark >= 70)
{  printf("1st class\n");
   printf("well done\n");
}
else if (mark >= 60)
     printf("2:1\n");
else if (mark >= 50)
     printf("2:2\n");
else if (mark >= 40)
     printf("3rd class");
else
printf("fail");
getch();
return 0;
}
```

Note that we left out the curly braces in some of these statements; when only a single statement is used the braces are not necessary, but two or more statements should be

enclosed in braces to indicate their association with an *if*, *while/for loop* or *function*. Only a single outcome will be evaluated. As soon as an if expression is evaluated as true, its statements will be executed and no other expression will be tested. The final else statements deals with all other outcomes that have not been met by any earlier tests.

fgets() is a general-purpose text input function in the C programming language, Run the following C code to learn this function.

```
#include <stdio.h>
#include <conio.h>
int main()
{
  char name[10];
  printf("What is your name?\n");
  fgets(name,10,stdin);
  printf("I am glad to meet you, %s",name);
  getch();
  return(0);
}
```

The C library function ***atoi()*** converts the string argument to an integer. Run the following code to understand this function.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int main() {
  char numchar[10];   //define a character variable
  int numnum;          //define a integer variable
  printf("Enter a number\n");
  fgets(numchar,10,stdin);    //enter a string
  numnum=atoi(numchar);   //covert string to integer
  printf("Entered number is %d",numnum);
  getch();
  return(0);
}
```

Part I: Personal Income Tax

1. First of all, study the examples above. Then write a similar program that allows the user to type in the personal annual income (**use *fgets* and *atoi***) and then prints the income groups using the if-else statements according to the table below.

Annual Income	Income groups
0 - 20,000	Low
20,001 – 100,000	Lower-middle
100,001 – 200,000	Upper-middle
200,001 –	High

The annual income lower than 0 is impossible. Please print an error message if the input is lower than 0.
Then, calculate the personal income tax according to the table below:

Income	Marginal Tax Rate
0 - 20,000	0
20,001 – 100,000	10%
100,001 – 200,000	20%
200,001-	30%

Please note this is the marginal tax rate. This means, if the income is 250,000, then the tax should be calculated as : (100,000-20,000)*10% + (200,000-100,000)*20% + (250,000-200,000)*30%.

Part II: Mathematical Practice with C Language

2. Ask user to enter values *a* and *b*, then calculate and displaya^b and *y* values.
$$y = tan^{-1}(a^b)$$

3. In the Gregorian calendar three criteria must be taken into account to identify leap years:
The year can be evenly divided by 4;
If the year can be evenly divided by 100, it is NOT a leap year, unless;
The year is also evenly divisible by 400. Then it is a leap year.

   Write a c program to check whether a year is leap year or not.

4. Ask user to enter two angles *x* and *y* in radians, then show the z value
$$z = ln^x + log_{10}^y, when\ x > y > 0$$
$$z = \sin(x) + \cos(y)\ , when\ x < y$$

Submission Requirement

(1) This practical has 4 problems. Please prepare one source file (.c file) per problem.

(2) Make sure that each source file can be compiled without error, and the corresponding executable file can produce the expected result.

(3) You can either submit 4 individual source files, or compress the source files in a zip file (or a rar file) and submit the compressed file.