Subject: PRF192- PFC
Workshop 01

Objectives:

(1) Reviewing for number systems
(2) Exploring memory of a C program

Recommendations
Part 1: Students do exercises using notebooks
Part 2: Students develop programs, run them, write down their memory structure to notebooks.

Part 1: Number systems

Exercise 1 (2 marks): Convert decimal numbers to binary ones

Decimal	4-bit Binary	Decimal	8-bit Binary	Decimal	16-bit Binary
9	1001	7	0000 0111	255	0000 0000 1111 1111
7		34		192	
2		125		188	
15		157		312	
12		162		517	
11		37		264	
6		66		543	
5		77		819	
8		88		1027	
13		99		2055	
14		109		63	

Exercise 2(2 marks): Convert decimal numbers to binary and hexadecimal ones

Decimal	Binary	Hexa.	Decimal	16-bit Binary	Hexadecimal
9	1001	9	255	0000 0000 1111 1111	00FF
127	0111 1111	9F	192		

125			188		
157			312		
162			517		
37			264		
66			543		
77			819		
88			1027		
99			2055		
109			63		

Exercise 3(2 marks): Compute
(b: binary, q: octal, h: hexadecimal)

3245q + 247q = ?q = ?b
1A7Bh + 26FE7h = ?h = ?b
1101101101b - 10110111b =?b
3654q – 337q =?q = ?b
3AB7h – 1FAh = ?h = ?b
36Ah – 576q = ? h = ?b
64AEh – 1001101b= ? q

101101111 b
\+ 100111011 b
110110001 b
110001101b

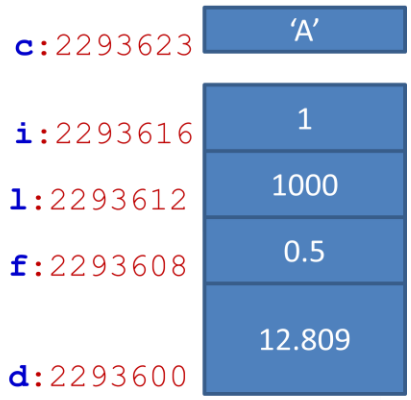
1011010 b* 1011b
1101000b + 2AB h + 345 q = ? h = ? q
3AFh / 1Ch =? b = ?d
3ACh – 562q = ?b = ? d
3FFA h / 327q = ?b = ? d

Exercise 4 (2 marks)

1- Show binary formats of 1-byte unsigned numbers: 251 , 163, 117
2- Show binary formats of 2-byte unsigned numbers: 551 , 160, 443
3- Show binary formats of 1-byte signed numbers: -51 , -163, -117, 320
4- Show the decimal values of 1-byte unsigned representations: :
01100011 b , 10001111 b , 11001010 b , 01001100 b

Part 2: Explore memory structure of programs

Sample



Vars_demo.c

```
/* Variables Demo - Operator &: address of */
#include <stdio.h>
#include <conio.h>
int main() {
   char c='A'; int i=1; long l=1000;
   float f=0.5f; double d=12.809 ;
   printf("Variable c: at addr: %u, value: %c, size: %d\n", &c, c, sizeof(c));
   printf("Variable i: at addr: %u, value: %d, size: %d\n", &i, i, sizeof(i));
   printf("Variable l: at addr: %u, value: %ld, size: %d\n", &l, l, sizeof(l));
   printf("Variable f: at addr: %u, value: %f, size: %d\n", &f, f, sizeof(f));
   printf("Variable d: at addr: %u, value: %lf, size: %d\n", &d, d, sizeof(d));
   getch();
}
```

```
G:\GiangDay\FU\PFC\PFC_Lab\Vars_demo.exe
Variable c: at addr: 2293623, value: A, size: 1
Variable i: at addr: 2293616, value: 1, size: 4
Variable l: at addr: 2293612, value: 1000, size: 4
Variable f: at addr: 2293608, value: 0.500000, size: 4
Variable d: at addr: 2293600, value: 12.809000, size: 8
```

Complete the code of following program then draw it's memory structure (2 marks)

```
#include <stdio.h>
int n;
double x;
char c1;
int main()
{   int m;
    short s;
    long L;
    float y;
    printf("Code of main:%u\n", &main));
    printf("Variable n, add:%u, memory size:%d\n", &n, sizeof(n));
    /* Your code to view address and memory size of other variables*/
    /* Complete the program, compile and run it */
    /* Draw the memory of the program*/
    getchar();
    return 0;
}
```