**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 | 0111 | 34 | 0010 0010 | 192 | 0000 0000 1100 0000 |
| 2 | 0010 | 125 | 0111 1101 | 188 | 0000 0000 1011 1100 |
| 15 | 1111 | 157 | 1001 1101 | 312 | 0000 0001 0011 1010 |
| 12 | 1100 | 162 | 1010 0010 | 517 | 0000 0010 0000 0101 |
| 11 | 1011 | 37 | 0010 0100 | 264 | 0000 0001 0000 1000 |
| 6 | 0110 | 66 | 0100 0010 | 543 | 0000 0010 0001 1111 |
| 5 | 0101 | 77 | 0100 1101 | 819 | 0000 0011 0010 1001 |
| 8 | 1000 | 88 | 0101 1000 | 1027 | 0000 0100 0000 0011 |
| 13 | 1101 | 99 | 0110 0011 | 2055 | 0000 1000 0000 0111 |
| 14 | 1110 | 109 | 0110 1011 | 63 | 0000 0000 0100 0001 |

**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 | 0000 0000 1100 0000 | 00F0 |
| 125 | 0111 1101 | 7D | 188 | 0000 0000 1011 1100 | BC |
| 157 | 1001 1101 | 9D | 312 | 0000 0001 0011 1010 | 13A |
| 162 | 1010 0010 | A2 | 517 | 0000 0010 0000 0101 | 205 |
| 37 | 0010 0100 | 24 | 264 | 0000 0001 0000 1000 | 108 |
| 66 | 0100 0010 | 42 | 543 | 0000 0010 0001 1111 | 21F |
| 77 | 0100 1101 | 4D | 819 | 0000 0011 0010 1001 | 329 |
| 88 | 0101 1000 | 58 | 1027 | 0000 0100 0000 0011 | 403 |
| 99 | 0110 0011 | 63 | 2055 | 0000 1000 0000 0111 | 807 |
| 109 | 0110 1011 | 6B | 63 | 0000 0000 0100 0001 | 41 |

**Exercise 3(2 marks): Compute**

(b: binary, q: octal, h: hexadecimal)

**3245q + 247q = 3492q = 3492b**

**1A7Bh + 26FE7h = 1A792h = 110010100110010b**

**1101101101b - 10110111b =100010110b**

**3654q – 337q =3317q = 3317b**

**3AB7h – 1FAh = 38E5hh = 1111000100101b**

**36Ah – 576q =D90h = -1101100100b**

**64AEh – 1001101b= 37061 q**

101101111 b

+ 100111011 b

110110001 b

110001101b

1011010 b\* 1011b = 1101001110b

1101000b + 2AB h + 345 q = 3E5 h = 997 q

3AFh / 1Ch =100001 b = 32d

3ACh – 562q = 11110110b = 254d

3FFA h / 327q = 110010b = 50d

**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

Result:

1. 251: 11111011b

163: 10100011b

117: 01110101b

1. 551: 00101001 00100111b

160: 00000000 10100000b

443: 00000001 10110111b

1. -51: 1101101b

-163: 10100001b

-117: 10001011b

320: range of signed 1 byte

1. 01100011b: 99

10001111b: 143

11001010b: 202

01001100b:76

**Part 2: Explore memory structure of programs**

**Sample**

**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));

printf("Variable x, add:%u, memory size:%d\n",&x, sizeof(x));

printf("Variable c1, add:%u, memory size:%d\n",&c1, sizeof(c1));

printf("Variable m, add:%u, memory size:%d\n",&m, sizeof(m));

printf("Variable s, add:%u, memory size:%d\n",&s, sizeof(s));

printf("Variable L, add:%u, memory size:%d\n",&L, sizeof(L));

printf("Variable y, add:%u, memory size:%d\n",&y, sizeof(y));

getchar();

return 0;

}