**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 | 0110 1101 | 188 | 0000 0000 1011 1100 |
| 15 | 1111 | 157 | 1001 1101 | 312 | 0000 0001 0011 1000 |
| 12 | 1100 | 162 | 1010 0010 | 517 | 0000 0010 0000 0101 |
| 11 | 1011 | 37 | 0010 0101 | 264 | 0000 0001 0000 1000 |
| 6 | 110 | 66 | 0100 0010 | 543 | 0000 0010 0001 1111 |
| 5 | 101 | 77 | 0100 1111 | 819 | 0000 0011 0011 0011 |
| 8 | 1000 | 88 | 0101 1000 | 1027 | 0000 0100 0000 0011 |
| 13 | 1101 | 99 | 0110 0010 | 2055 | 0000 1000 0000 0111 |
| 14 | 1110 | 109 | 0110 1101 | 63 | 0000 0000 0011 1111 |

16-32-48-64-80-96-112-128-144

**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 | 7F | 192 | 0000 0000 1100 0000 | 00C0 |
| 125 | 0110 1101 | 6D | 188 | 0000 0000 1011 1100 | 00BC |
| 157 | 1001 1101 | 9D | 312 | 0000 0001 0011 1000 | 0138 |
| 162 | 1010 0010 | A2 | 517 | 0000 0010 0000 0101 | 0205 |
| 37 | 0010 0101 | 25 | 264 | 0000 0001 0000 1000 | 0108 |
| 66 | 0100 0010 | 42 | 543 | 0000 0010 0001 1111 | 021F |
| 77 | 0100 1111 | 4F | 819 | 0000 0011 0011 0011 | 0333 |
| 88 | 0101 1000 | 58 | 1027 | 0000 0100 0000 0011 | 0403 |
| 99 | 0110 0010 | 62 | 2055 | 0000 1000 0000 0111 | 0807 |
| 109 | 0110 1101 | 6D | 63 | 0000 0000 0011 1111 | 003F |

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

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

**3245q + 247q = 1701 + 167 = 3514q = 0111 0100 1100b**

**1A7Bh + 26FE7h = 6779 + 159719 = 28A62h = 0010 1000 1010 0110 0010b**

**1101101101b - 10110111b =877 – 183 = 0010 1011 0110b**

**3654q – 337q = 3315q = 0110 1100 1101b**

**3AB7h – 1FAh = 38BDh = 0011 1000 1011 1101b**

**36Ah – 576q = 1ECh = 0001 1110 1100b**

**64AEh – 1001101b= 62141q**

101101111 b

+ 100111011 b

110110001 b

110001101 b

10111101000 b

1011010 b\* 1011b = 0011 1101 1110b

1101000b + 2AB h + 345 q = 3F8h = 1770q

3AFh / 1Ch = 0010 0001b = 33d

3ACh – 562q = 0010 0011 1010b = 570d

3FFA h / 327q = 0100 1100b = 76d

**Exercise 4 (2 marks)**

1. Show binary formats of 1-byte unsigned numbers:

251: 1111 1011b

163: 1010 0011b

117: 0111 0101b

1. Show binary formats of 2-byte unsigned numbers:

551: 0000 0010 0010 0111b

160: 0000 0000 1010 0000b

443: 0000 0001 1011 1011b

1. Show binary formats of 1-byte signed numbers:

-51: 1100 1101b

-163:

-117: 1000 1011b

320:

1. Show the decimal values of 1-byte unsigned representations: :

01100011 b = 99d

10001111 b = 143d

11001010 b = 202d

01001100 b = 76d

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

**Sample**

**Complete the code of following program then draw it’s memory structure**

**(2 marks)**

|  |  |  |
| --- | --- | --- |
|  | **add** | **size** |
| **n** | **4223024** | **4** |
| **x** | **4223032** | **8** |
| **c1** | **4223040** | **1** |
| **m** | **6487580** | **4** |
| **s** | **6487578** | **2** |
| **L** | **6487572** | **4** |
| **y** | **6487568** | **4** |



#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;

}