# C PROGRAMING TRAINING COURSE

**HUMAX VIETNAM R&D**

Tutor: Vu Quang Trong | Henry

*version: 1.0 R, updated on: 2017 Feb 8th*

# Day 3: Structure & Linked List, Dynamic Memory Test

Test Date: **17-Feb-2017**

Total Tests: **04 questions**

Maximum Score: **100 points**

Total Time: **120 mins**

Modify the code to make it work properly as required. Code is written in C language, C99 standard, and compiled under Linux.

---

## Test 1

**Score: 20**

A register can be declared with a field of bits. Write funtions to turn on/off a bit in a register

```
#define BIT0 0x01
#define BIT1 0x02
#define BIT2 0x04
#define BIT3 0x08
#define BIT4 0x10
#define BIT5 0x20
#define BIT6 0x40
#define BIT7 0x80

union _register {
    struct _bits {
        unsigned char BIT_7:1;
        unsigned char BIT_6:1;
        unsigned char BIT_5:1;
        unsigned char BIT_4:1;
        unsigned char BIT_3:1;
        unsigned char BIT_2:1;
        unsigned char BIT_1:1;
        unsigned char BIT_0:1;
    } bits;
    unsigned char value;
};

void setVaue(union _register *reg, unsigned char bit, unsigned char value)
{
    // your code goes here
}

int main()
{
    union _register REG_A;
    // your code goes here to test implemented function
    setVaue(&REG_A, BIT1, 0);
```

```
    return 0;
}
```

# Test 2

**Score: 40**

Create a linked list with basic functions: create, insert, remove, delete

A node is declared as below:

```
struct _node {
    int value;
    struct _node *next;
};
```

Now create a list of nodes.

```
struct _node * create();
int  delete(struct _node * list);
void insert(struct _node * list, int value);  // if value is already in the list, skip inserting, if not, append new
void remove(struct _node * list, int value);  // remove all instances of value
```

# Test 3

**Score: 20**

How to detect a cycle in a linked list?

```
int detectCycledList(struct _node * list)
{
    // return 1 if list has a cycle, otherwise, return 0
}
```

# Test 4

**Score: 20**

Write a function to scan free memory. To avoid crash of your computer in case you program doesnt work properly, run your app using `ulimit` tool as below:

```
# ulimit -v 64m
# ./yourapp
# ulimit -v unlimited
```

Your code might use `malloc` and `free`

```
int getAvalableMemory()
{
    int ret = 0;
    // your code goes here
    return ret;
}
```

---

End of the test.

*this document is written in GitHub Markdown format*