Test - C for embedded systems

The following test is a simulation of questions used in a Professional Interview for a Real-Time \Embedded Engineer position; the questions will focus on the C language (Not RT Concepts).

The purpose of this test is to help you prepare for the Professional Interviews and determine the level of your knowledge in C for embedded systems.

Using the test we can find out more accurately which subjects you will need to strengthen and achieve the goal of preparing you to master the Science of RT\Embedded Engineering.

The subjects you will be questioned on are:

1. C for embedded systems
2. Basic Computing

The time to complete the test is 2 hours

Good luck

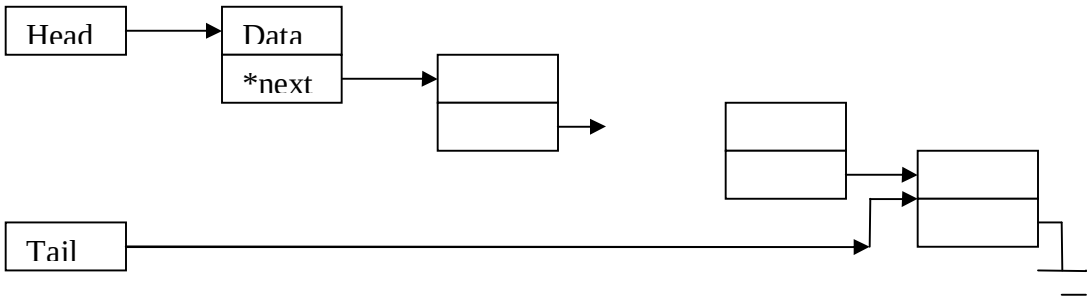
Question 1:

```
Typedef struct
{
        unsigned int m_type;
        unsigned int m_lParam;
} Message;

typedef struct Q;

void InitQ(Q* pQ, unsigned int size);
void PushQ(Q* pQ, Message message);
Message PopQ(Q*pQ);

TaskA()
{
        Message msg;

        While (1)
        {
                msg = PopQ(g_pQ);
                ...
        }
}

foo()
{
        char text[] = "New member";
        Message msg;

        ...

        msg.m_type          = MSG_LOG;
        msg.m_lParam        = (const char*)text;

        PushQ(g_pQ, msg);

        ...
}

TaskB()
{
        ...
        foo();
        ...
}
```

1) מה הבעיה בקטע הקוד?
2) הצע פתרון לבעיה.

Question 2

1. You are given the following linked list with N elements :



2. write the Appropriate structure declaring the Linked list(Data is of type int).
3. Write code to build a Linked list with N elements, ask the user to input the Data variable .
4. write code to sort the linked list using the data Variable from the largest element first to the smallest last.
5. Write code to reverse the Linked list so, Head points to the last element and Tail to the first.
6. Please explain how you would recognize a loop in a Linked List?

Question 3

A. Why do we need C? Why not make do with assembly? (Elaborate main difference between them).

B. What is the difference between the "const" key word and `#define`? Which one is better?

C. What is the difference between passing *by ref* and *by val*? Point out pros/cones Of each method.

D. Given the source code below, in the following table, fill in where each variable is stored while the program is at run time:

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>..
#include <iostream>
#define MAX_DOG_AGE  14
//-------------------------------------------
typedef struct Dog
  {
  unsigned char name[100];
  unsigned int age;

  }Dog;
//-------------------------------------------
void displayDogAge(Dog dog_)
{
     printf("%d",dog_.age);
}
//-------------------------------------------
void displayDogName(Dog* pdog)
{
     Dog* dog_pointer = pdog;
     printf("%s",dog_pointer->name);
}
//-------------------------------------------

Dog doggy;
const int some_age =7;

//-------------------------------------------

int main ()
{

     Dog rexi = {"rexi",4};
```

```
        char* some_name = "rexi";

        displayDogAge(rexi);
        printf("\n");
        displayDogName(&rexi);

        return 0;
}
>>>
```

Question 4

A Hardware system reads packets into a large buffer.
All packets are fixed in size.
Each packet includes 32 bit Header and 1k Data fields .
Header fields:

SOF –Start of Frame 8 bit.
SID - Subscriber I.D 12 bit
CC- Continuity Check 4 bit.

The Packets are part of a streamed VOD and have to synchronized per SID, to ensure that the CC field is used. The CC field should increment by 1 for 2 consecutive packets per SID.

SOF	
SID	
SID	not used
not used	CC
1K DATA	

1. Write the appropriate structure for each packet.

2. Write a function which verifies the Continuity Check, the function should recognize missing packets between 2 consecutive packets with the same SID and store the number of missing packets per SID.

3. Point out the system's limitations.