Lab 2 , Embedded C

1- Define a structure for Car models for a factory, where the structure has the Car ID, Model name, Max speed, battery level and weight.

2- Define array of structures from the same type Car, then scan the information from the user for the chosen array in main( ) -[or in a function]-.

3- Write a C function to print information of a chosen structure of the array of structures

Function prototype:

```
void print_info( struct Car info[ ], int index);
```

```
/* define new struct called New_Car with the same info as the previous */
#include <stdio.h>
#include <stdlib.h>
#include <gotoxy.h>
struct Car
{
   int ID;
   char Name[10];
   int Max_speed;
   int Battery_level;
   int Weight;
};
struct Car c[3];
int main()
{
   char ch;
   int i;
   int j;

   do
   {
      system("cls");
      gotoxy(25,2);
      printf("Please, Enter the index \n");
      gotoxy(30,4);
      scanf("%d", &i);
      gotoxy(25,6);
      printf("Car no. %d \n", i);
      gotoxy(10,10);
      printf(" ID:  \n");
      gotoxy(10,12);
      printf(" Name: \n");
      gotoxy(10,14);
      printf(" Max_speed: \n");
      gotoxy(10,16);
      printf(" Battery_level: \n");
      gotoxy(10,18);
      printf(" Weight: \n");
```

```
        gotoxy(30,10);
        scanf ("%d", &c[i].ID);
        _flushall();
        gotoxy(30,12);
        scanf ("%s", &c[i].Name);
        _flushall();
        gotoxy(30,14);
        scanf ("%d", &c[i].Max_speed);
        _flushall();
        gotoxy(30,16);
        scanf ("%d", &c[i].Battery_level);
        _flushall();
        gotoxy(30,18);
        scanf ("%d", &c[i].Weight);
        _flushall();
        printf("\n");
        printf("\n");

        printf("Do you want to continue?, y or n! \n");
        ch = getch();
    }
    while (ch == 'y');
    printf("Enter the no of the car you want. \n");
    scanf("%d",&j);
    Car_info(c,j);
    return 0;
}

void Car_info( struct Car c[ ], int i)
{
    printf("ID: %d \n", c[i].ID);
    printf("\n");
    printf("Name: %s \n", c[i].Name);
    printf("\n");
    printf("Max_speed: %d \n", c[i].Max_speed);
    printf("\n");
    printf("Battery_level: %d \n", c[i].Battery_level);
    printf("\n");
    printf("Weight: %d \n", c[i].Weight);
}
```

****
****

4- Write a C function to assign information to structure using a function that takes the data you want to assign as parameters and return a struct.

struct New_Car assign_data (int id, char name[10], char mx_speed, float battery ,double Kg)

```
#include <stdio.h>
#include <stdlib.h>
#include <gotoxy.h>

struct Car
{
   int ID;
   char Name[10];
   int Max_speed;
   int Battery_level;
   int Weight;
} c;
struct Car assign_data (int id, char *name, char mx_speed, float battery,double Kg);

int main()
{
   struct Car c;

   c=assign_data(1,"samaa",4,4,4);
   gotoxy(10,10);
   printf(" ID:  %d \n", c.ID);
   gotoxy(10,12);
   printf(" Name: %s\n",c.Name);
   gotoxy(10,14);
   printf(" Max_speed: %d\n",c.Max_speed);
   gotoxy(10,16);
   printf(" Battery_level: %d \n", c.Battery_level);
   gotoxy(10,18);
   printf(" Weight: %d \n", c.Weight);
}
struct Car assign_data (int id, char* name, char mx_speed, float battery,double Kg)
{
   struct Car mycar;
   mycar.ID=id;
   int i = 0;
   while(name[i] != '\0')
   {
      mycar.Name[i] = name[i];
      i++;
   }
   mycar.Name[i] = '\0';
   mycar.Max_speed=mx_speed;
   mycar.Battery_level=battery;
   mycar.Weight=Kg;
   return mycar;
}
```

****
****

4.a-Write a C function to assign information to structure using a function that takes the pointer to the struct and assign some info to it, assume the data and assign it in the function itself.

void assign_data_w_ptr(struct new_car *car_ptr)

```
#include <stdio.h>
#include <stdlib.h>
#include <gotoxy.h>

struct New_Car
{
   int ID;
   char Name[10];
   int Max_speed;
   int Battery_level;
   int Weight;
};

int main()
{
   struct New_Car car;
   gotoxy(10,10);
   printf(" ID:  \n");
   gotoxy(10,12);
   printf(" Name: \n");
   gotoxy(10,14);
   printf(" Max_speed: \n");
   gotoxy(10,16);
   printf(" Battery_level: \n");
   gotoxy(10,18);
   printf(" Weight: \n");
   Assign_data_w_ptr(&car);
   return 0;
}

void Assign_data_w_ptr(struct New_Car *car_ptr)
{
   gotoxy(30,10);
   scanf ("%d", &car_ptr->ID);
   _flushall();
   gotoxy(30,12);
   scanf ("%s", &car_ptr->Name);
   _flushall();
   gotoxy(30,14);
   scanf ("%d", &car_ptr->Max_speed);
   _flushall();
   gotoxy(30,16);
   scanf ("%d", &car_ptr->Battery_level);
```

```
    _flushall();
    gotoxy(30,18);
    scanf ("%d", &car_ptr->Weight);
}
```

****

5- How to find out if your PC is big endian or small endian.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char x=1;
    if (x&1)
        printf("Little Endian. \n");
    else
        printf("Big Endian. \n");

    return 0;
}
```

****

6- Write a function that prints "HELLO ITI" every second without using C "Delay" function, assuming one second event occur every 120 Million Cycles on an intel 7 CPU. And return from the function the state, whether it printed something or not

```
#include <stdio.h>
#include <stdlib.h>

void Delay()
{
    while(1)
    {
    for (int i=0; i<120e6 ; i++) {}
    printf("HELLO ITI \n");
    }
}

int main()
{
    Delay();
    return 0;
}
```