Program 1:

```
//Toni Hunter 187009925
//ASSIGNMENT 6 PROGRAM 1
// a)Define a structure called "part" containing unsigned int variable "partNumber" and char array "partName" with values that may be as long as 25 characters (including the terminating null character).
// b)Define Part to be a synonym for the type struct part.
// c)Use Part to declare variable a to be of type struct part, array b[10] to be of type struct part and variable ptr to be of type pointer to struct part.
// d)Read a part number and a part name from the keyboard into the individual members of variable a.
// e)Assign the member values of variable a to element 3 of array b.
// f)Assign the address of array b to the pointer variable ptr.
// g)Print the member values of element 3 of array b using the variable ptr and the structure pointer operator to refer to the members.
//
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct part
{
   unsigned int partNumber;
   char partName[25]; //25 characters long
};

typedef struct part Part; // synonym for struct part

int main()
{
   Part a, b[10], *ptr;

   printf("Please enter part number: ");
   scanf("%u", &a.partNumber);
   printf("Please enter part name: ");
   scanf("%s", &a.partName);

   b[3] = a;
   ptr = b;

   printf("\nPart number: %u", ptr[3].partNumber);
   printf("\nPart name: %s\n", ptr[3].partName);
}
```

Program 2:

```
//Toni Hunter 187009925
//ASSIGNMENT 6 PROGRAM 2
// Reverse the order of an unsigned integer's bits. value should be given from user.
// print user value and reversed value.
//
#include <stdio.h>
#include <stdlib.h>

//protoypes
unsigned reverseBits(unsigned num);
void displayBits(unsigned num);

int main()
{
   unsigned a; //user input variable

   printf("Please enter an integer: ");
   scanf("%u", &a); // unsigned user scan

   printf("\nBelow are the ORIGINAL bits: \n");
   displayBits(a);

   a = reverseBits(a);
   printf("\nBelow are the REVERSED bits:\n");
   displayBits(a);

   return 0;
}

unsigned reverseBits(unsigned num)
{
   unsigned mask = 1;
   unsigned renum = 0;

   for(int i=0; i<=15; i++)
   {
      renum <<= 1; //shift left by 1
      renum |= (num & mask);
      num >>= 1; //shift right by 1
   }
   return renum;
}

void displayBits(unsigned num)
{
   unsigned displayMask = 1 << 15; //define and left shift 15 bits
   printf("%7u = ",num);
```

```
    for(unsigned int c=1; c<=16; c++) //loops through bits
    {
        putchar(num & displayMask ? '1' : '0');
        num <<= 1; //shift num left by 1

        if(c%8==0) //output space after 8 bits
        {
            putchar(' ');
        }
    }
    putchar('\n');

}
```