Question 1

Source code:

```
#include<stdio.h>


int add(int x,int y){
		return x+y;
	}
int subtract(int x, int y){
		return x-y;
	}
int multiply(int x, int y){
		return x*y;
	}
int divide(int x, int y){
		return x/y;
	}
int main(){

	char operator;
	int x, y;
	int result = 0;
    printf("WELCOME TO SIMPLE CALCULATOR\n");
   printf("----------------------------\n");
	printf("Enter an operator..... \n");
	printf("Enter + for addition: \n");
	printf("Enter - for subtraction: \n");
	printf("Enter * for multiplication: \n");
	printf("Enter / for division: \n");
	scanf("%c", &operator);

	printf("Enter two operands: \n");
	scanf("%d %d",&x, &y);
 switch(operator)
 {
	case '+':
		result = add(x,y);
		break;
	case '-':
		result = subtract(x,y);
		break;
	case '*':
		result = multiply(x,y);
		break;
	case '/':
		result = divide(x,y);
		break;
	// operator doesn't match if the operator number is wrong
	default:
		printf("Error! operator is not correct");
	}
```

```
            /* Print the result */
    printf("%d %c %.d = %.d\n", x, operator, y, result);
          return 0;
          }
```

Output:

```
WELCOME TO SIMPLE CALCULATOR
----------------------------
Enter an operator.....
Enter + for addition:
Enter - for subtraction:
Enter * for multiplication:
Enter / for division:
/
Enter two operands:
10
2
10 / 2 = 5
```

Question 2

Source code:

```
#include<stdio.h>
int main(){
	int ID;
	int password;

	printf("Please enter your ID here: \n");
	scanf("%d",&ID);

	switch(ID){
		case 12190102:
		printf("Now enter your password: \n");
		scanf("%d", &password);

		switch(password){
			case 123:
			printf("ugyen you are welcomed in GCIT.\n");
			break;

			default:
			printf("Try again!! your password is wrong.\n");
			break;
		}
		break;
		case 12190106:
			printf("Enter your password: \n");
			scanf("%d",&password);
			switch (password){
				case 000:
					printf("WELCOME yeshi dema\n");
					break;

				default:
					printf("Incorrect Password. Try again!!\n");
					break;
		}
		break;

		default:
		printf("Try again!! Your ID is wrong.\n");
		break;

	}

}
```

Output:

Please enter your ID here:
12190102
Now enter your password:
000
Try again!! your password is wrong.

Another output:

Please enter your ID here:
12190106
Enter your password:
000
WELCOME yeshi dema

Question 3

Source code:

```
#include<stdio.h>

int convertBinary(int n)
{
   static int biNo,r,fctor = 1;

   if(n != 0)
   {
       r = n % 2;
       biNo = biNo + r * fctor;
       fctor = fctor * 10;
       convertBinary(n / 2);//calling the function convertBinary itself recursively
   }
   return biNo;
}
int main()
{
   int biNo;
   int n;

   printf("\n\n Recursion : Converting number to binary :\n");
   printf("-------------------------------------------------\n");

   printf(" Input any  number : ");
   scanf("%d",&n);

   biNo = convertBinary(n);//call the function convertBinary
   printf(" The Binary value of the given no. %d is : %d\n\n",n,biNo);
   return 0;
}
```

Output:

```
Recursion : Converting number to binary :
-------------------------------------------------
 Input any  number : 50
 The Binary value of the given no. 50 is : 110010
```

Question 4

Source code:

```
#include <stdio.h>

#define N   10

int main( void )
{
   int a[N] = {6,1,4,3,7,8,2,10,9,11};
   int i, j;

   printf( "Array[%d] = { ", N );
   i = 0;
   do
   {
      printf( "%d", a[i] );
   } while ( ++i < N && printf( ", " ) );
   printf( " };\n");

   i = 0; j = N;
   while ( i != j )
   {
      if ( a[i] % 2 == 0 )
      {
         ++i;
      }
      else if ( a[--j] % 2 == 0 )
      {
         int tmp = a[j];
         a[j] = a[i];
         a[i] = tmp;
      }
   }

   printf( "Array[%d] = { ", N );
   i = 0;
   do
   {
      printf( "%d", a[i] );
   } while ( ++i < N && printf( ", " ) );
   printf( " };\n");

   return 0;
}
```

Output:

```
Array[10] = { 6, 1, 4, 3, 7, 8, 2, 10, 9, 11 };
Array[10] = { 6, 10, 4, 2, 8, 7, 3, 1, 9, 11 };
```

Question 5

a) There is error in line number 4 since the variable p doesn't get the address of s therefore the variable p cannot access the string indexes. But if we make the variable p as a pointer and while printing we can increment the value of p by 1 and make the whole as a pointer than the output can be displayed without an error.

Correct Code is:

```
#include <stdio.h>
void main()
{
char *s= "hello";
char *p = s;
printf("%c%c", *(p+1), s[1]);
}
```

Output:
he

b) Error since the break statement is not within the loop therefor there is an error. We usually use break to terminate the program in loops and switch cases so here in the if-else statement we use the keyword called exit() which will terminate the recursion. And we also do have to include C program library

Correct Code is:

```
# include <stdio.h>
# include <stdlib.h>
int main( )
{
int i = 0 ;
i++ ;
if ( i <= 5 )
{
printf ( "C is very easy\n" ) ;
exit(1);
main();
}
return 0 ;
}
```

Output:
C is very easy

c) First of all the value of 'i' is 0. There is a function named val which initializes the value of 'i' as 100 and prints the value of 'i' as 100 and after printing it increments the value of 'i'.
In line number 6 when we print the value of 'i' in the main method we get the value as 0 since in the main method the value of 'i' is 0. after that in line number 7 we call the void val function in the main method so it will print the value of I as 100 since the value of 'i' in the val function is 100 initially. Now the value of 'i' has been incremented so when we print the value of 'i' from the main method we get the incremented value of 'i' which is 101. And finally in line 9 we again print the

value of ‘i’ from the val function we get the initial value of ‘i’ from the val function we get it as 100 since it is the initial value of ‘i’ in the val function.