

Birla Institute of Technology & Science, Pilani Hyderabad Campus

Second Semester 2020-2021

Computer Programming [CS F111]

Lab 8

Q1. Write a program to convert a Binary number into its equivalent Decimal number. Do not use Arrays or any other data structures.

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int convert (long long n);
5  int main ()
6  {
7      long long n;
8      printf ("Enter a binary number: ");
9      scanf ("%lld", &n);
10     printf ("%lld in binary = %d in decimal", n, convert (n));
11     return 0;
12 }
13
14 int convert (long long n)
15 {
16     int dec = 0, i = 0, rem;
17     while (n != 0) {
18         rem = n % 10;
19         n /= 10;
20         dec += rem * pow (2, i);
21         ++i;
22     }
23     return dec;
24 }
```

Output:

```
Enter a binary number: 11001
11001 in binary = 25 in decimal
```

Q2. Write a program to convert a DECIMAL number into its equivalent OCTAL number. Do not use Arrays or any other data structures.

```
1  #include <stdio.h>
2  #include <math.h>
3  long long convert(int n);
4  int main() {
5      int n;
6      printf("Enter a decimal number: ");
7      scanf("%d", &n);
8      printf("%d in decimal = %lld in octal", n, convert(n));
9      return 0;
10 }
11
12 long long convert(int n) {
13     long long bin = 0;
14     int rem, i = 1, step = 1;
15     while (n != 0) {
16         rem = n % 8;
17         n /= 8;
18         bin += rem * i;
19         i *= 10;
20     }
21     return bin;
22 }
```

Output1:

```
Enter a decimal number: 12
12 in decimal = 14 in octal
```

Output2:

```
Enter a decimal number: 337
337 in decimal = 521 in octal

...Program finished with exit code 0
Press ENTER to exit console.
```

Q3. Write a program to find out 1's complement of a Decimal integer. Do not use Arrays or any other data structures.

```
1  #include <stdio.h>
2  long long convert(long long int n) {
3      long long bin = 0;
4      long long int rem, i = 1, step = 1;
5      while (n != 0) {
6          rem = n % 2;
7          n /= 2;
8          bin += rem * i;
9          i *= 10;
10     }
11     return bin;
12 }
13 int main() {
14     long long int num1, num2 = 5, no, counter;
15     long long fab = 1, count = 0;
16     int power = 1, rem;
17     printf("Enter a decimal integer:");
18     scanf("%lld", &no);
19     num1 = convert(no);
20     printf("Binary equivalent:%lld\n", num1);
21     while (num1 != 0) {
22         counter = num1 % 10;
23         if (counter == 0)
24             num2 = num2 * 10 + 1;
25         else
26             num2 *= 10;
27         num1 /= 10;
28     }
29     while (num2 % 10 != 5) {
30         rem = num2 % 10;
31         num1 = num1 * 10 + rem;
32         num2 /= 10;
33     }
34     printf("1's Complement with leading 0s' suppressed: %lld", num1);
35 }
```

Output:

```
Enter a decimal integer:23
Binary equivalent:10111
1's Complement with leading 0s' suppressed: 1000
```

(**Note:** As integers are used, the binary output gets shortened, because leading zeroes will have no significance for an int, e.g. 01000 will be output as 1000, which is acceptable).

Q4. Write a program to do binary addition. Take two binary numbers as input and add them to get their binary sum. Take your input in long long int format. Do not use Arrays or any other data structures.

```
1  #include <stdio.h>
2  #include <math.h>
3  long long convert(long long int n){
4      int dec = 0, i = 0, rem;
5      while (n != 0) {
6          rem = n % 10;
7          n /= 10;
8          dec += rem * pow(2, i);
9          ++i;
10     }
11     return dec;
12 }
13 long long convertB(long long int n) {
14     long long bin = 0;
15     long long int rem, i = 1, step = 1;
16     while (n != 0) {
17         rem = n % 2;
18         n /= 2;
19         bin += rem * i;
20         i *= 10;
21     }
22     return bin;
23 }
24 void main() {
25     long long int num1, num2 = 5, no, counter, fab = 1, count = 0;
26     printf("Enter both the numbers:\n");
27     scanf("%lld %lld", &num1, &num2);
28     num1 = convert(num1);
29     num2 = convert(num2);
30     no = num1 + num2;
31
32     printf("Sum: %lld", convertB(no));
33 }
34 }
```

Output:

```
Enter both the numbers:
1101
1101
Sum: 11010
```

TASKS:

1. Write a program to PRINT the hexadecimal equivalent of a input decimal number. Do not use any arrays.

2. Take 2 binary numbers as input and subtract the second number from the first one. For this find the 1's complement of the second number and add it to the first number. The steps are as follows. [Do not use Arrays or any other data structures]

i. Find the 1's complement of the second number.

ii. Add it to the first number.

iii. If the result of addition has a carry over then it is ignored and an 1 is added in the last bit of the result.

iv. If there is no carry over, then 1's complement of the result of addition is obtained to get the final result and it is negative.