Perserose P. Catalan

CMSC 21 Assignment No.2

1.

as1.c as2.c as3.c

```c
#include <stdio.h>

int main (void){

    /*declare variables*/

    int num, n1, rev_num;

    /*ask to input a number to reverse*/
    printf("Please enter a 2-digit number: ");
    scanf("%d",&num);


    /*We get the last digit of the number
      by getting the remainder and store it
      in a variable*/
    n1 = num%10;

    /*This will remove the last digit from
      the number*/
    num = num/10;


    /*Multiply the remainder by 10 and add
      the new value of the variable num
      to get the reverse*/

    rev_num = (n1*10) + num;

    /*print result*/
    printf("Reverse: %d ",rev_num);

    return 0;

}
```

EXAMPLE OUTPUT:

```
Please enter a 2-digit number: 27
Reverse: 72
Process returned 12 (0xC)   execution time : 6.083 s
Press any key to continue.
```

2.

as1.c as2.c as3.c

```c
#include <stdio.h>

int main (void)
{

    /*declare variables*/

    int num, n1, n2, rev_num;

    /*ask to input a number to reverse*/
    printf("Please enter a 3-digit number: ");
    scanf("%d",&num);


    /*We get the last digit of the number
      by getting the remainder and store it
      in a variable*/
    n1 = num%10;
    num = num/10;
    n2 = num%10;
    num = num/10;


    /*Multiply the remainder by 100 and 10 respectively
      and add the new value of the variable num
      to get the reverse*/

    rev_num = ((n1*100) + (n2*10) + num);

    /*print result*/
    printf("Reverse: %d \n",rev_num);

    return 0;
}
```

EXAMPLE OUTPUT:

```
Please enter a 3-digit number: 258
Reverse: 852
Process returned 13 (0xD)   execution time : 4.794 s
Press any key to continue.
```

3.

```c
#include <stdio.h>

int main (void){

    int i, j, k;

    i = 3;j = 4; k = 5;
    printf ("%d\n", i < j || ++j < k);

    i = 7; j = 8; k = 9;
    printf("%d\n", i - 7 && j++ < k);

    i = 7; j = 8; k = 9;
    printf("%d\n", (i = j) || (j == k));
    printf("%d %d %d\n", i, j, k);

    i = j = k = 1;
    printf ("%d\n", ++i || ++j && ++k) ;
    printf ("%d %d %d\n", i, j, k);

    return 0;

}
```

OUTPUT:

```
1
0
1
8 8 9
1
2 1 1
```