

HW1

1. Write a program to print Binary representation of a given number (unsigned integer 16 bit).

- Main idea:

Step 1: if number > 1

Push number on stack.

Recursively call function with number = number / 2.

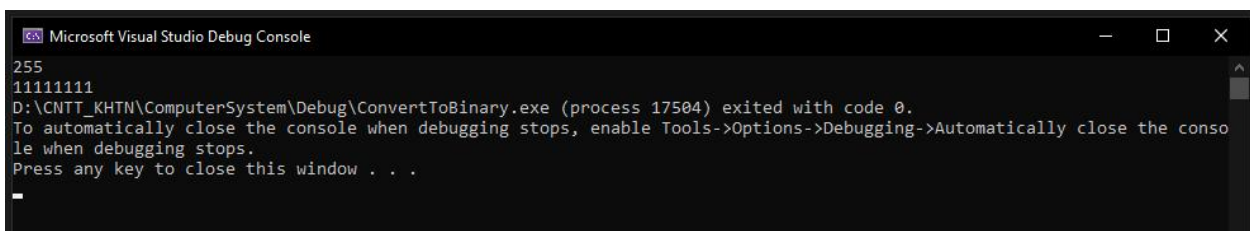
Step 2:

Pop number from stack.

- Source code:

```
1  #include<iostream>
2
3  using namespace std;
4
5  void convertToBinary(unsigned int num)
6  {
7      if (num > 1)
8          convertToBinary(num / 2);
9
10     cout << num % 2;
11 }
12
13 int main()
14 {
15     int num;
16     cin >> num;
17     convertToBinary(num);
18     return 0;
19 }
```

- Output screen:



```
Microsoft Visual Studio Debug Console
255
11111111
D:\CNTT_KHTN\ComputerSystem\Debug\ConvertToBinary.exe (process 17504) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

2. Compile the program in the file constant_limit2.c and answer to the following questions.

- The main difference between these pairs of lines.
- The difference has some meaning.

```
printf("The maximum value of UNSIGNED INT = %u\n", UINT_MAX);  
printf("The maximum value of UNSIGNED INT = %d\n", UINT_MAX);
```

printf ("The maximum value of UNSIGNED INT = %u\n", UINT_MAX);	printf ("The maximum value of UNSIGNED INT = %d\n", UINT_MAX);
%u: unsigned int	%d: signed int
Meaning: print the maximum value of unsigned int	Meaning: print the signed value of UINT_MAX
Result: The maximum value of UNSIGNED INT = 4294967295	Result: The maximum value of UNSIGNED INT = -1
Explain: The value of UINT_MAX will be 32 numbers 111..111. Type %u will read UINT_MAX as unsigned int. So, it will print $2^{31} + 2^{30} + \dots + 2^0 = 4294967295$	Explain: The value of UINT_MAX will be 32 numbers 111..111. Type %d will read UINT_MAX as signed int. So, it will calculate by two's complement representation. 111...111 (32) Sign bit is 1 -> negative number Absolute value is the complement of $111...111 + 1$ -> $00...000 + 1$ -> 1 (decimal) The integer is -1. So, it will print -1

```
printf("The maximum value of UNSIGNED LONG = %ld\n", ULONG_MAX);
printf("The maximum value of UNSIGNED LONG = %u\n", ULONG_MAX);
```

printf ("The maximum value of UNSIGNED LONG = %ld\n", ULONG_MAX);	printf ("The maximum value of UNSIGNED LONG = %u\n", ULONG_MAX);
%ld: long int	%u: unsigned int
Meaning: print the signed value of ULONG_MAX	Meaning: print the maximum value of unsigned int
Result: The maximum value of UNSIGNED LONG = -1	Result: The maximum value of UNSIGNED LONG = 4294967295
<p>Explain: The value of ULONG_MAX will be 32 numbers 111..111. Type %ld will read ULONG_MAX as long int (signed). So, it will calculate by two's complement representation. 111...111 (32) Sign bit is 1 -> negative number Absolute value is the complement of 11...111 + 1 -> 00...000 + 1 -> 1 (decimal) The integer is -1. So, it will print -1</p>	<p>Explain: The value of ULONG_MAX will be 32 numbers 111..111. Type %u will read ULONG_MAX as unsigned int. So, it will print $2^{31} + 2^{30} + \dots + 2^0 = 4294967295$</p>