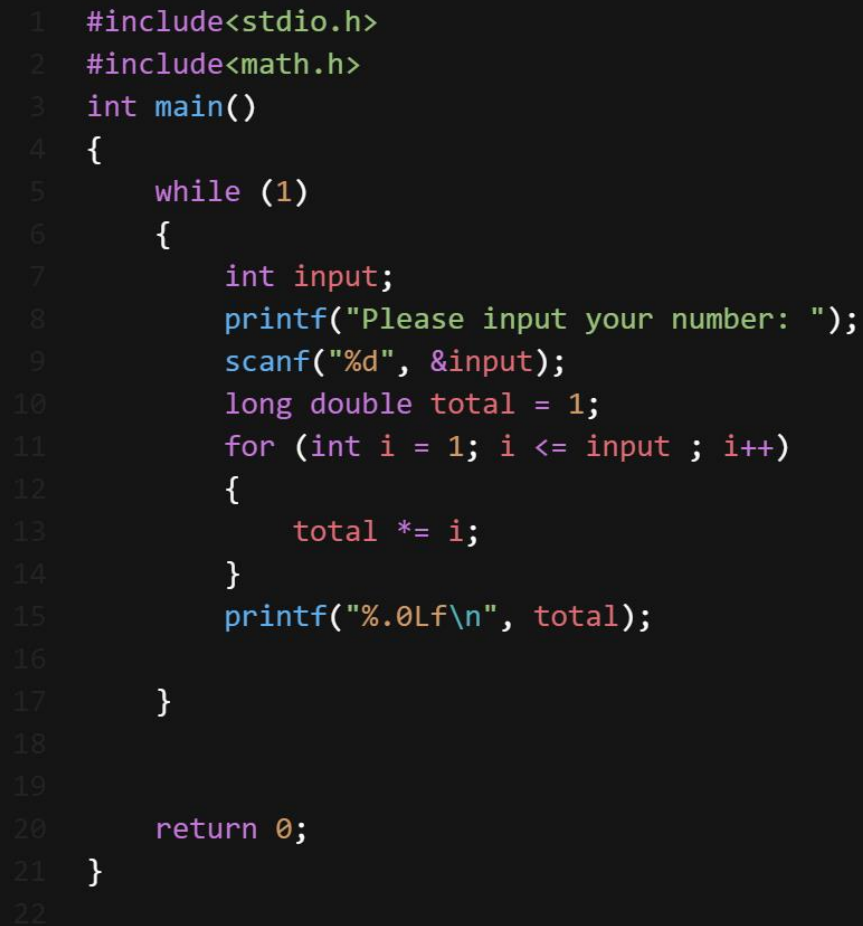


Program Factoria;

- Logic

1) V1:



```
1  #include<stdio.h>
2  #include<math.h>
3  int main()
4  {
5      while (1)
6      {
7          int input;
8          printf("Please input your number: ");
9          scanf("%d", &input);
10         long double total = 1;
11         for (int i = 1; i <= input ; i++)
12         {
13             total *= i;
14         }
15         printf("%.0Lf\n", total);
16     }
17
18
19
20     return 0;
21 }
22
```

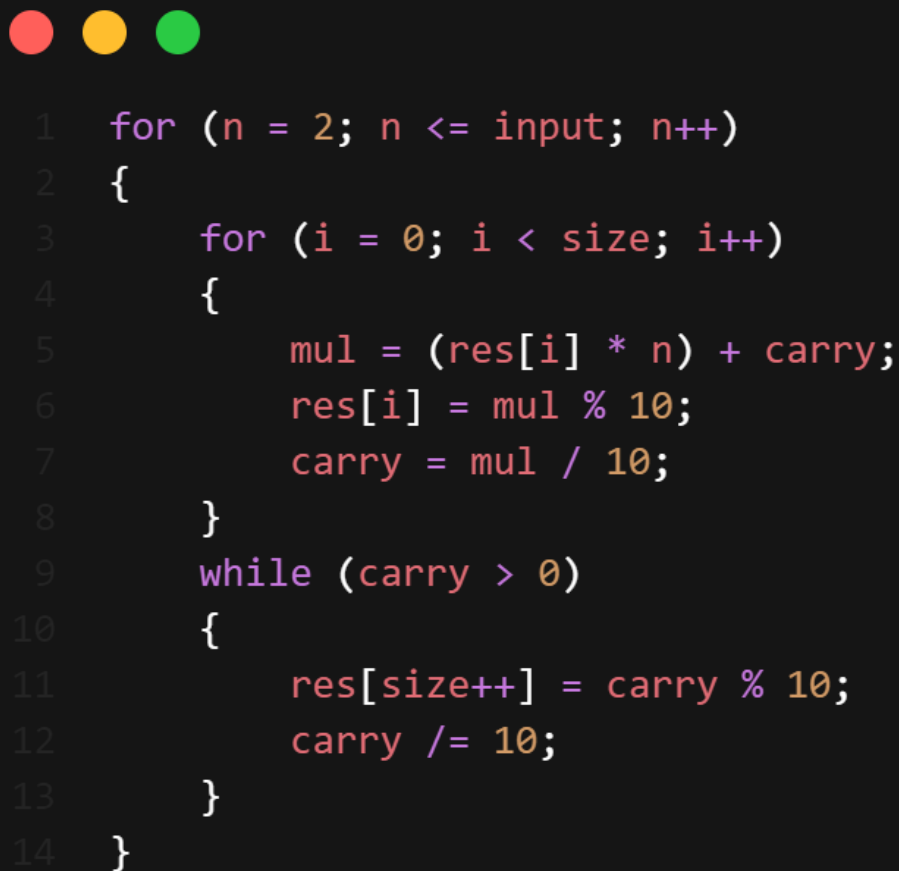
ຫຼັກການທຳງານເບື້ອງຕົ້ນຈະປັນການວິນ loop ໃຫ້ຄູນຈຳນວນຕັ້ງແຕ່ 1 ຈົນຮອດຄ່າທີ່ປ້ອນໃສ່ໂດຍການ ເກັບຄ່າທີ່ໄດ້ໃສ່ໃນ total ແລ້ວຄູນເຂົ້າກັນໄປເລື້ອຍໆ ໂດຍ total ຈະໃຊ້ເປັນ long double ເພື່ອເກັບຄ່າທີ່ມີ ຈຳນວນຫຼາຍ (ຈາກທີ່ເຮັດມາສາມາດ factorial ຮອດພຽງແຕ່ 1754.

2) V2:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int input, carry, size, mul,n,i;
6      int res[10000];
7
8      printf("Please Enter your number: ");
9      scanf("%d", &input);
10
11     res[0] = 1;
12     carry = 0;
13     size = 1;
14
15     for (n = 2; n <= input; n++)
16     {
17         for (i = 0; i < size; i++)
18         {
19             mul = (res[i] * n) + carry;
20             res[i] = mul % 10;
21             carry = mul / 10;
22         }
23         printf("%d %d\n", carry,i);
24         while (carry > 0)
25         {
26
27             res[size++] = carry % 10;
28             carry /= 10;
29         }
30         printf("w %d %d\n", carry,size);
31     }
32
33     printf("\nFactorial = ");      // Printing the result
34     for (i = size - 1; i >= 0; i--) // Array is stored in reverse so printing in reverse
35         printf("%d", res[i]);
36     return 0;
37 }
38
```

ທຳອິດແມ່ນສ້າງ array ສຳລັບເກັບຄ່າທີ່ໄດ້ ໂດຍໄດ້ຕັ້ງຂະໜາດໄວ້ທີ່ 10000. ຕໍ່ມາຈະກຳນົດຄ່າ `res[0] = 1` ເພື່ອໃຫ້ຕຳແໜ່ງທີ່ 1 ໃນ array ເທົ່າ 1 ກຳນົດ `size = 1` ໂຕນີ້ໝາຍເຖິງຂະໜາດທີ່ຢູ່ໃນ array.

ຈາກ loop ແມ່ນ loop ສຳລັບການຄຳນວນ factorial ໂດຍຄ່າເລີ່ມຕົ້ນແມ່ນຈະເລີ່ມທີ່ 2 ວິນຮອບໄປຈົນເຖິງຄ່າທີ່ input ເຂົ້າມາ ໂດຍທາງໃນ loop ຈະມີການທຳງານຂອງ loop ອີກ 2 ອັນຄື for, while.



```
1  for (n = 2; n <= input; n++)
2  {
3      for (i = 0; i < size; i++)
4      {
5          mul = (res[i] * n) + carry;
6          res[i] = mul % 10;
7          carry = mul / 10;
8      }
9      while (carry > 0)
10     {
11         res[size++] = carry % 10;
12         carry /= 10;
13     }
14 }
```

Loop for ຈະເປັນການຄຳນວນ factorial ໂດຍຈະໃຊ້ `mul` ມາເກັບຄ່າທີ່ຄຳນວນໄວ້ ຈາກນັ້ນຈະເອົາໄປເກັບໄວ້ໃນ array `res`. `res[i] = mul % 10` ຈະເປັນການຫານເອົາເສດເພື່ອທີ່ຈະເອົາຄ່າໄປຄິດໄລ່ຕໍ່.



```
1  for (i = 0; i < size; i++)
2  {
3      mul = (res[i] * n) + carry;
4      res[i] = mul % 10;
5      carry = mul / 10;
6  }
```

ໃນສ່ວນຂອງ carry ຄ້າຍໆຄືກັບຕົວທີ່ສິ່ງຄ່າໄປໃນແຕ່ລະບ່ອນໃນ array ຈາກ loop while ຖ້າເມື່ອໃດທີ່ $carry > 0$ ມັນຈະທຳການນຳເອົາຄ່າທີ່ຢູ່ໃນ $carry \% 10$ ແລ້ວເອົາໄປໄວ້ໃນ array $res[i]$ ຈາກນັ້ນຈະຫານໃຫ້ 10 ເພື່ອ reset ຄ່າເປັນ 0 ເພື່ອອອກຈາກ loop



```
1  while (carry > 0)
2  {
3      res[size++] = carry % 10;
4      carry /= 10;
5  }
```

Loop ສຸດທ້າຍຈະເປັນການ loop ເອົາຄ່າທີ່ຢູ່ໃນ array ອອກມາສະແດງ



```
1  for (i = size - 1; i >= 0; i--)
2      printf("%d", res[i]);
```