

Лабораторна робота №13

Створення програм з використанням рекурсивних функцій

Мета: навчитися складати й реалізовувати алгоритми та програми з використанням рекурсивних функцій мовою C++.

Обладнання: ПК.

Програмне забезпечення: Середовище C++.

Завдання:

1. Набрати код програми і перевірити її роботу.
Обчислення факторіалу за допомогою рекурсії.

```
// ваше прізвище і група
#include <iostream.h>
#include <conio.h>
long int factorial(int i)
{
    if (i==0) return 1;
    else return i*factorial(i-1);
}
int main()
{
    int n;
    int i;
    int res;
    cin >> n;
    cout << factorial(n);
    getch();
}
```

2. Відсортувати одномірний масив **a[n]** за зростанням елементів, використовуючи рекурсивну функцію швидкого сортування.

```
// ваше прізвище і група
#include <iostream.h>
#include <conio.h>
template <class T>
void QuickSortRecur(T* a, int Nend)
{
    // вихідний масив a[], a[Nend] - його останній елемент
    int i = 0, j = Nend;
    T rab, c;
    c = a[Nend >>1]; /* вибір центрального елемента з використанням операції побитового зсуву вправо */
    do
    { while (a[i] < c) i++;
      while (a[j] > c) j--;
      if (i <= j)
```

```

{ rab = a[i];
  a[i++] = a[j];
a[j--] = rab; }
}
while (i <= j);
/* рекурсивні виклики QuickSortRecur(), якщо є, що сортувати */
if (j > 0) QuickSortRecur(a, j);
if (Nend > i) QuickSortRecur(a+i, Nend-i);
}
void main()
{ int a[] = {2, 7, 6, 9, 45, 4, 3, 67, 104, 1, 99, 72, 43, 8, 4, 28,
100};
  int n = sizeof(a)/sizeof(int);
QuickSortRecur(a, n-1);
cout << "\n Result Quick-sortirovki massiva" << endl;
for (int i = 0; i<n; i++)
cout << a[i] <<' ';
getch();
}

```

3. Вивести на екран усі зростаючі послідовності довжиною **k**, елементами яких є натуральні числа від **1** до **n**. Передбачається, що **k** не перевищує **n**, інакше таких послідовностей не існує.

```

// ваше прізвище і група
#include <iostream.h>
#include <conio.h>
using namespace std;
const int m=20;
/* максимальна розмірність масиву зростаючої послідовності */
int t=1, n, k, a[m];
/* k – довжина зростаючої послідовності
n – довжина відрізка натуральних чисел, з яких формуються послідовності
generate() - рекурсивна функція генерації зростаючих послідовностей */
int generate()
{
int i;
if (t == k) // одна з послідовностей повністю сформована
{
// виведення послідовності
for (i = 0; i<k; i++)
cout << a[i] <<' ';
cout << endl;
}
else // формування послідовності
{ t++;
for (i = a[t-2] +1; i <= t-k+n; i++)

```

```
{ a[t-1] = i;
generate(); // рекурсивний виклик функції generate()
}
t--;
}
}
int main()
{ int j;
cout << "input k, n (k<n)\n k=";
cin >> k;
cout << "n=";
cin >> n;
cout <<endl;
for (j = 1; j<=n-k+1; j++)
{a[t-1]=j;
generate();
}
getch();
}
```