

**ТЕМА: ВВЕДЕНИЕ В ЯЗЫК
ПРОГРАММИРОВАНИЯ «C++»****Домашнее задание 2****ЗАДАНИЕ 1**

Вывести на экран один куплет любимого стихотворения или песни, с указанием автора или исполнителя. Используйте escape-последовательности для форматирования.

Подсказка 1

Если у Вас возникли сложности с выполнением данного задания, нажмите кнопку «Подсказка». С полным решением задания вы сможете ознакомиться, нажав на кнопку «Решение»

Решение 1**ЗАДАНИЕ 2**

Вывести на экран стих «*Every hunter wants...*» таким образом, чтобы каждый «цвет» начинался с новой строки и соответствующим количеством табуляций.

Подсказка 2

Если у Вас возникли сложности с выполнением данного задания, нажмите кнопку «Подсказка». С полным решением задания вы сможете ознакомиться, нажав на кнопку «Решение»

Решение 2**ЗАДАНИЕ 3**

Напишите объявление о продаже чего-нибудь и выведите его на экран, в том виде как оно должно было бы быть напечатано. ►

Например:

Продам щенков.
Порода: ротвейлер.

...

Тел.: 222-22-22

222 | 222 |

22 | 22 |

22 | 22 |

и т.д.

Рисунок 1

Подсказка 3

Если у Вас возникли сложности с выполнением данного задания, нажмите кнопку «Подсказка». С полным решением задания вы сможете ознакомиться, нажав на кнопку «Решение»

Решение 3

ПОДСКАЗКА К ЗАДАНИЮ 1

1. Какая команда используется для вывода информации в консоль?
2. Какой управляющий символ (escape-последовательность) используется для перехода на следующую строку?
3. В каком месте строки с командой вывода текста в консоль нужно разместить escape-последовательность для перехода на следующую строку.
4. Сколько нужно строк кода и команд `cout <<` для вывода, например, пяти строк стихотворения?

ПОДСКАЗКА К ЗАДАНИЮ 2

1. Какая команда используется для вывода информации в консоль?
2. Какой управляющий символ (escape-последовательность) используется для перехода на следующую строку?
3. В каком месте строки с командой вывода текста в консоль нужно разместить escape-последовательность для перехода на следующую строку?
4. Какой управляющий символ (escape-последовательность) используется для табуляции (отступа)? А если необходимо сделать несколько отступов?
5. В каком месте строки с командой вывода текста в консоль нужно разместить escape-последовательность для табуляции (отступа)?

ПОДСКАЗКА К ЗАДАНИЮ 3

1. Какая команда используется для вывода информации в консоль?
2. Какой управляющий символ (escape-последовательность) используется для перехода на следующую строку?
3. В каком месте строки с командой вывода текста в консоль нужно разместить escape-последовательность для перехода на следующую строку?
4. Какой управляющий символ (escape-последовательность) используется для табуляции (отступа)? А если необходимо сделать несколько отступов?
5. В каком месте строки с командой вывода текста в консоль нужно разместить escape-последовательность для табуляции (отступа)?
6. Если величина отступа в начале строки одинакова для всех строк текста, как это повлияет на количество символов escape-последовательности в этих строках?
7. Может ли использоваться символ escape-последовательности (например, табуляции) внутри текстового сообщения?

РЕШЕНИЕ ЗАДАНИЯ 1

Управляющие символы (или как их ещё называют — escape-последовательность) — символы, которые выводятся в поток вывода (с помощью команды `cout`), с целью форматирования вывода или печати некоторых управляющих знаков C++.

Все управляющие символы при использовании в коде обрамляются двойными кавычками, если в консоль необходимо вывести какое-то сообщение, то управляющие символы можно записывать сразу в необходимом месте сообщения.

Задача состоит в последовательном (т.е. одна за другой) выводе строк (так как куплет — это набор строк, каждая из которых начинается с новой строки).

Из управляющих символов (escape-последовательности) нам понадобится символ перехода на следующую строку (`\n`) для каждой следующей строки куплета стихотворения.

Пусть необходимо вывести следующий куплет стихотворения (8 строк текста):

```
Hold fast to dreams
For if dreams die
Life is a broken-winged bird
That cannot fly.
Hold fast to dreams
For when dreams go
Life is a barren field
Frozen with snow.
```

1. Выводим строку в консоль с описанием названия программы (по желанию, данная строка кода может быть пропущена) с помощью команды `cout` (находится в библиотеке

`iostream`, которая должна быть обязательно подключена перед функцией `int main()` и оператора вывода `<<`. Для вывода следующей информации с новой строки используем символ перехода на следующую строку (`\n`) в конце.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.1\n\n";
}
```

Внутри сообщения `"Home task #2.1\n"` мы использовали escape-последовательность `«\n»` для того, чтобы после вывода текста `"Home task"` программа перешла на новую строку, и следующая запись в консоли (первая строка куплета) соответственно выводилась с новой строки.

2. Создаем столько строк кода с командами `cout` и escape-последовательностью `«\n»` в конце сообщения (текст одной строки стихотворения), сколько строк есть в куплете стихотворения (8 в нашем примере)

Т. е. для вывода, например, первой строки куплета необходимо:

```
cout << "Hold fast to dreams\n";
```

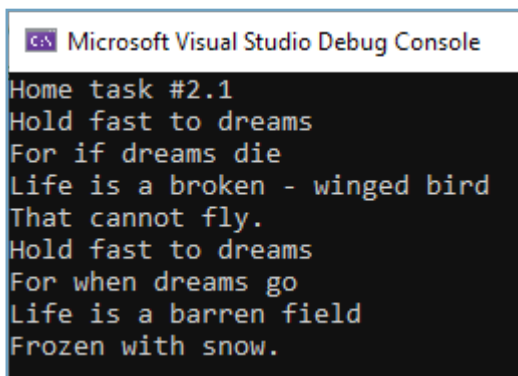
Первая часть сообщения (текста в кавычках) содержит строку куплета (*Hold fast to dreams*) и завершается escape-последовательностью `«\n»`, т.е. и нужный текст, и escape-последовательность находятся в одних (общих) кавычках.

Остальные строки кода, которые обеспечивают вывод следующих строк куплета, реализованы таким же образом.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.1\n\n";
    cout << "Hold fast to dreams\n";
    cout << "For if dreams die\n";
    cout << "Life is a broken - winged bird\n";
    cout << "That cannot fly.\n";
    cout << "Hold fast to dreams\n";
    cout << "For when dreams go\n";
    cout << "Life is a barren field\n";
    cout << "Frozen with snow.\n";
    return 0;
}
```

Результаты работы программы (в консоли):



```
Microsoft Visual Studio Debug Console

Home task #2.1
Hold fast to dreams
For if dreams die
Life is a broken - winged bird
That cannot fly.
Hold fast to dreams
For when dreams go
Life is a barren field
Frozen with snow.
```

Рисунок 2

РЕШЕНИЕ ЗАДАНИЯ 2

Задача состоит в последовательном (т. е. одна за другой) выводе строк, каждая из которых содержит одно слово из предложения *«Every hunter wants to know where does pheasant sit»*.

Сдвиг строки (слова) вправо осуществляется за счет табуляции (один символ табуляции обеспечивает один сдвиг вправо).

Из управляющих символов (escape-последовательности) нам понадобится символ перехода на следующую строку (`\n`) для каждой следующей строки (слова) и символ горизонтальной табуляции (`\t`) для сдвига текста вправо.

1. Выводим строку в консоль с описанием названия программы с помощью команды `cout` (находится в библиотеке `iostream`, которая должна быть обязательно подключена перед функцией `int main()`) и оператора вывода `<<`. Для вывода следующей информации с новой строки используем символ перехода на следующую строку (`\n`) в конце.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.2\n\n";
```

Внутри сообщения *«Home task #2.2\n»* мы использовали escape-последовательность *«\n»* для того, чтобы после вывода текста `Home task` программа перешла на новую строку и следующая запись в консоли (первое слово `Every`

предложения «*Every hunter wants to know where does pheasant sit*») соответственно выводилось с новой строки.

2. Создаем столько строк кода с командами `std::cout` и escape-последовательностями «`\t`» в начале и «`\n`» в конце сообщения, сколько строк необходимо вывести (7 в нашем примере).

2.1. Первая строка (слово `Every`) не должна сдвигаться, поэтому escape-последовательность «`\t`» не помещается в начало текста:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.2\n\n";
    cout << "Every" << "\n";
```

Однако следующее слово (`hunter`) должно быть выведено с новой строки, поэтому в конце дополнительно выводится escape-последовательность «`\n`» после оператора вывода `<<`.

- 2.2. Вторая строка сдвинута вправо на одну табуляцию, поэтому нам нужно в начало текстового фрагмента после первого оператора вывода `<<` поместить escape-последовательность «`\t`».

```
#include <iostream>
using namespace std;
```

```
int main()
{
    cout << "Home task #2.2\n\n";
    cout << "Every" << "\n";
    cout << "\tHunter" << "\n";
}
```

Аналогично предыдущему шагу для вывода, следующего слово (**Wants**) с новой строки в конце дополнительно выводится escape-последовательность «**\n**» после второго оператора вывода **<<**.

- 2.3. Третья строка сдвинута вправо уже на две табуляции, поэтому нам нужно в начало текстового фрагмента после первого оператора вывода **<<** поместить две escape-последовательности «**\t**».

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.2\n\n";
    cout << "Every" << "\n";
    cout << "\tHunter" << "\n";
    cout << "\t\tWants" << "\n";
}
```

Аналогично завершаем строку escape-последовательностью «**\n**» после второго оператора вывода **<<**.

- 2.4. Можно заметить следующую закономерность: количество табуляций перед словом на единицу меньше номера слова в предложении. Значит, для вывода чет-

вертого слова необходимо добавить в начало текстового фрагмента три символа табуляции, для вывода пятого слова — четыре символа табуляции и т. д.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.2\n\n";
    cout << "Every" << "\n";
    cout << "\tHunter" << "\n";
    cout << "\t\tWants" << "\n";
    cout << "\t\t\tTo know" << "\n";
    cout << "\t\t\t\tWhere" << "\n";
    cout << "\t\t\t\t\tDoes Pheasant" << "\n";
    cout << "\t\t\t\t\t\tSit" << "\n\n";
    return 0;
}
```

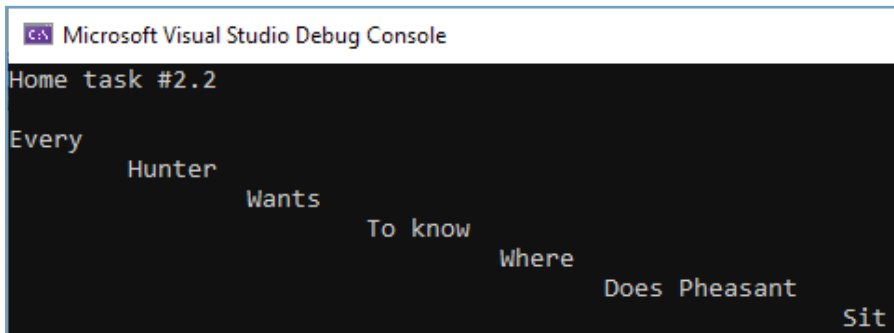
Общий код программы:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.2\n\n";
    cout << "Every" << "\n";
    cout << "\tHunter" << "\n";
    cout << "\t\tWants" << "\n";
    cout << "\t\t\tTo know" << "\n";
    cout << "\t\t\t\tWhere" << "\n";
```

```
cout << "\\t\\t\\t\\t\\tDoes Pheasant" << "\\n";  
cout << "\\t\\t\\t\\t\\t\\t\\t\\tSit" << "\\n\\n";  
return 0;  
  
}
```

Результаты работы программы (в консоли):



Microsoft Visual Studio Debug Console

Home task #2.2

Every
 Hunter
 Wants
 To know
 Where
 Does Pheasant
 Sit

Рисунок 3

РЕШЕНИЕ ЗАДАНИЯ 3

Задача состоит в последовательном (т. е. одна за другой) выводе строк, каждая из которых (в данном примере) представляет одно предложение.

Для обеспечения перехода на следующую строку после вывода информации в текущей будем использовать escape-последовательность (`\n`).

Для выравнивания текста по центру будем подбирать необходимое количество символов табуляции (`\t`).

1. Выводим строку в консоль с описанием названия программы (по желанию, данная строка кода может быть пропущена) с помощью команды `cout` (находится в библиотеке `iostream`, которая должна быть обязательно подключена перед функцией `int main()`) и оператора вывода `<<`. Для вывода следующей информации с новой строки используем символ перехода на следующую строку (`\n`) в конце.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.3\n\n";
}
```

2. Следующая строка объявления «Продам щенков.» выводится по центру, т.е. нам понадобится несколько символов табуляции перед текстом, сразу после первого оператора вывода `<<`.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.3\n\n";
    cout << "\t\t\t\tPuppies for sale" << "\n";
```

Следующая строка объявления должна быть выведена с новой строки, поэтому в конце дополнительно выводится escape-последовательность “\n” после оператора вывода <<.

3. Повторяем шаг 2 с двумя следующими предложениями объявления

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.3\n\n";
    cout << "\t\t\t\tPuppies for sale" << "\n";
    cout << "\t\t\t\tDog breed: Rottweiler" << "\n";
    cout << "\t\t\t\tTel: 222-22-22" << "\n";
```

4. Следующая строка имеет и внутренний отступ, т.е. мы будем использовать последовательность из символов табуляции не только в начале текстового фрагмента, но и в середине

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.3\n\n";
    cout << "\t\t\t\tPuppies for sale" << "\n";
    cout << "\t\t\t\tDog breed: Rottweiler" << "\n";
    cout << "\t\t\t\tTel: 222-22-22" << "\n";
    cout << "\t\t\t\t222\t|22-22|" << "\n";
}
```

5. Повторяем шаг 4 с необходимым текстовым фрагментом между символами |.....| нужное количество раз

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Home task #2.3\n\n";
    cout << "\t\t\t\tPuppies for sale" << "\n";
    cout << "\t\t\t\tDog breed: Rottweiler" << "\n";
    cout << "\t\t\t\tTel: 222-22-22" << "\n";
    cout << "\t\t\t\t222\t|22-22|" << "\n";
    cout << "\t\t\t\t222\t|22-22|" << "\n";
    cout << "\t\t\t\t222\t|22-22|" << "\n";
    return 0;
}
```


Результаты работы программы (в консоли):

Microsoft Visual Studio Debug Console

Home task #2.3

```
Puppies for sale
Dog breed: Rottweiler
Tel: 222-22-22
222 |22-22|
222 |22-22|
222 |22-22|
```

Рисунок 4