# Keywords, Identifier, Literals, Operators and Expression Assignment

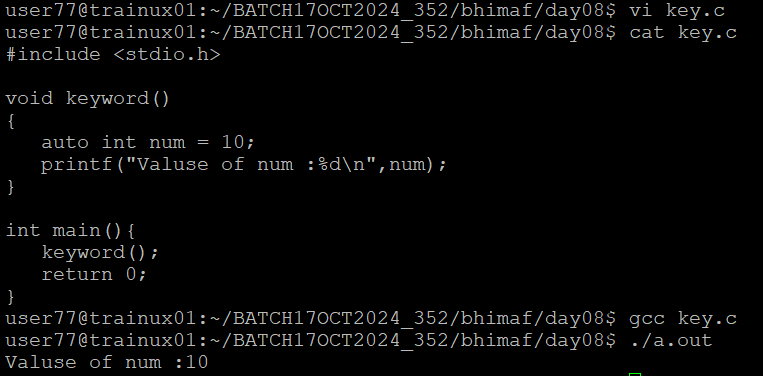
**Mandatory:**

1. Choose all valid identifiers
   1. int int
   2. int \_numvalue
   3. float price\_money
   4. char name1234567890123456789012345678901234567890
   5. char name value
   6. char $name

Ans) Valid options are b. int\_numvalue, float price\_money and remaining are invalid.

Technically valid but too long for C standard limit i.e option d.

1. What is the meaning of the following keywords, show the usage
   1. auto: keyword specifies that a variable has automatic storage duration.This means the variable is stored in memory only while the function in which it is defined is active.by default, all local variables inside functions are auto, so the auto keyword is rarely used explicitly.



* 1. extern: Declares in another file or in a different scope. This is used for linking purpose and allows a variable to be shared across multiple files.

A computer screen with white text

Description automatically generated

* 1. volatile: Tells the complier that the value of a variable can change at any time, often due to external factors such as hardware or signals.This prevents the complier from optimizing the code in a way that assumes the variables value won’t change unexpectedly

A computer screen with text and symbols

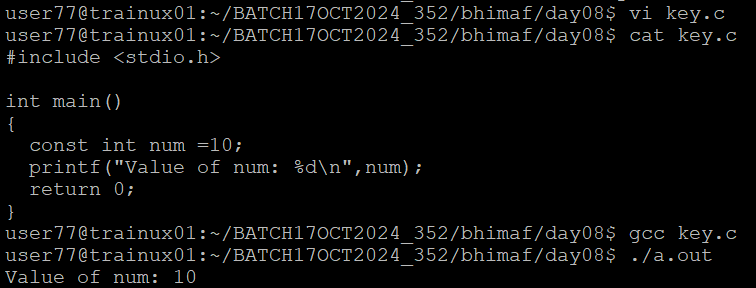
Description automatically generated

* 1. sizeof: Return the size of a variable or data type. Used to determine the amount of memory required for an array, structure, or specific data type.

A computer screen with white text

Description automatically generated

* 1. const: It makes a variable constant, meaning it cannot be modified after it’s initial assignment. Attempting a change a const variable’s value will result in a complier error



1. Explain the difference between the following variables.
   1. char \*ptr = “ABC”;

Ans) ptr is a pointer to a string literal “ABC”. The string “ABC” is stored in read-only memory, and ptr points to the address of the memory. This is string literal since “ABC“ is stored in read-only memory. However, modifying \*ptr is not allowed.

* 1. char arr[]=”ABC”;

Ans) arr is an array of characters initialized with the string “ABC”. The characters are copied into arr and arr itself has its own allocated memory, separate from read-only memory. The array arr is not string literal rather it contains a copy of the characters from the literals “ABC”, allowing modifications.

- Can you manipulate the contents of ptr? Why?

Ans) No, because it points to a read-only string literal. Attempting to modify “ABC” may causeundefined behaviour, often resulting in a segmentation fault.

-Can you manipulate the contents of arr? Why?

Ans) Yes, because it is modifiable character array.

-Which one of the above is a string literal?

Ans) ptr is a string literal.

1. Predict the output of the following code .

void main()

{

//set a and b both equal to 5.

int a=5, b=5;

//Print them and decrementing each time.

//Use postfix mode for a and prefix mode for b.

printf("\n%d %d",a--,--b);

printf("\n%d %d",b++,--b);

}

//output is

1. 4

3 4

1. Refer the code snippet. It fails with error. Fix it.

#include<stdio.h>

int main()

{

int i,k;

const int num;

/\* for(i = 0;i < 9;i++)

{

k = k + 1;

} \*/

num = num + k; /\* Compiler gives the error here \*/

printf("final value of k:%d\n",k);

printf("value of num:%d\n",num);

return 0;

}

Ans) const int num declaration: num is constant which means its value can’t be changed after initization. But trying to assign num=num+k; will cause a compiler error.

Uninitialized variables: int k and const int num are declared but not initialized which can lead to undefined behaviour.

So, initiakize k and num.

6. Consider the following code snippet. Evaluate the value of f1, f2 and f3.

int main()

{

int i = 10;

int j = 3;

float f1 = i / j;

float f2 = (float ) i / j;

float f3 = (float ) (i / j);

}

Ans) f1= 3.000000 f2= 3.333333 f3= 3.000000