Roll No: IT057

Name: Kathiriya Darshakkumar

C) Pick any programming language and find at least two compiler/interpreter for it. Now compare the two compilers and explain the benefits and drawbacks of each compiler .

Language: C language

Compilers:

* Turbo c
* GNU (GCC) (Windows MinGW)

1. **Turbo c**

* It is an old, non-standard C compiler.
* It can be used with any operating systems, such as Windows, Linux.
* It was developed by Borland.
* It has menus and mouse-based interaction facilities as well, which makes it Graphical User Interface IDE.
* Turbo's C++ 4.5 compiler do not support latest C++ standards.
* Initial Release in the year 1990.

1. **GNU (GCC) (Windows MinGW)**

* It is a primitive, standard C compiler.
* It is used and it comes mostly with Unix based Operating systems.

(also used with MinGW32or64 in windows system with Code::blocks ,VS code likes ide)

* It was developed by Free Software Foundation.
* Users need to use via a command-line interface.
* GNU Compiler supports C++14 and C++17.
* Initial Release in the year 1987.

**benefits and drawbacks**

1)

* Turbo C is obsolete, and does not support (and, AFAIK, does not even claim to support) recent enough language standards like [C11](https://en.wikipedia.org/wiki/C11_(C_standard_revision)) or C++11 or [C++14](https://en.wikipedia.org/wiki/C++14). The language accepted by TurboC is worse that what the recent language standards specify. Quite often, some code written for TurboC is not standard conforming, and won’t even compile on a standard conforming implementation (like GCC).
* latest version of [GCC](http://gcc.gnu.org/) is supporting (almost all of) recent standards like C11 or C++14.

2)

* TurboC is proprietary software; you cannot see its source code (legally).
* GCC is [free software](https://en.wikipedia.org/wiki/Free_software), you can download legally, study and improve its source code. Read [What is free software and why is it so important for society?](http://www.fsf.org/about/what-is-free-software) That fact alone makes me prefer GCC.

3)

* TurboC produces poor executable code (whose performance is not very good) GCC is able to optimize (when asked to, e.g. g++ -O2 …) and can produce executables having good performance.
* If you are using only the C language features of C89 then Turbo C is perfectly fine.
* If you want to use newer features of C, better shift to GCC.

4)

* The GCC compiler is a portable compiler,it runs on most platforms available today.
* Whereas a turbo c compiler is not a portable compiler .

5)

* Turbo C compiler supports far pointers whereas GCC compiler doesn't.

6)

* Turbo C is 16 bit compiler, so the size of int is 2 bytes, size of pointer is 2 bytes.
* In GCC, we have 32 bit and 64 bit compiler. The size of int is 4 bytes, size of pointer is 4 bytes in 32 bit and 8 bytes in 64 bit version.

1. Turbo C: getch(); defined in conio.h – that wait for a key or character to press.

GCC (all IDE window or linux): not required; screen display press any key to continue.

1. Turbo C: Filename should not exceed 8 characters. (space is not allowed in the file name)

GCC (All IDE window or linux):Filename name can exceed 8 characters,contains space. Maximum limit of filename depends on OS in use.

**Turbo c vs GCC (use MinGW - VS Code)**

Some new features are

1. Flexible variable declaration.

#include <stdio.h>

int main()

{

    int a , b;

    printf("enter two numbers : ");

    scanf(" %d %d",&a,&b);

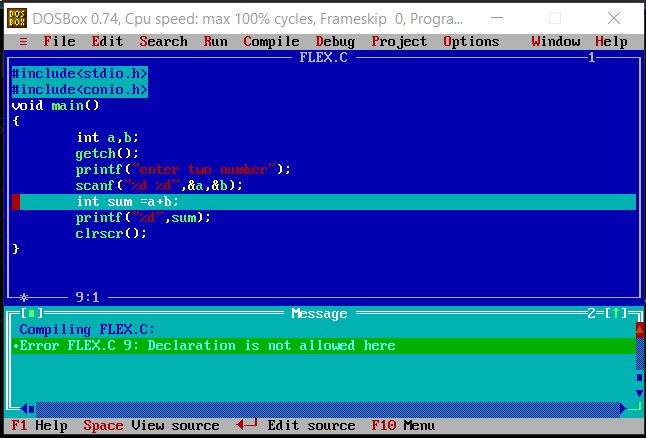
    int sum = a + b; //flexible variable declaration.

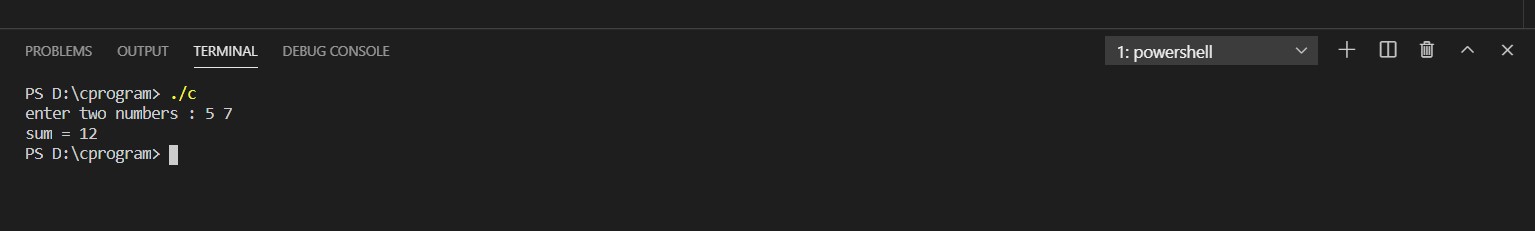
    printf("sum = %d \n",sum);

    return 0;

}

This code works in GCC(vs code) but does not work in turbo c





1. variable length array

#include <stdio.h>

void main()

{

int n;

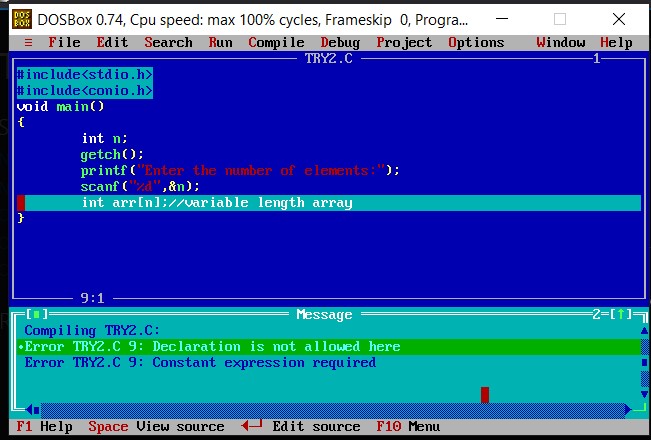
printf("Enter the number of elements : ");

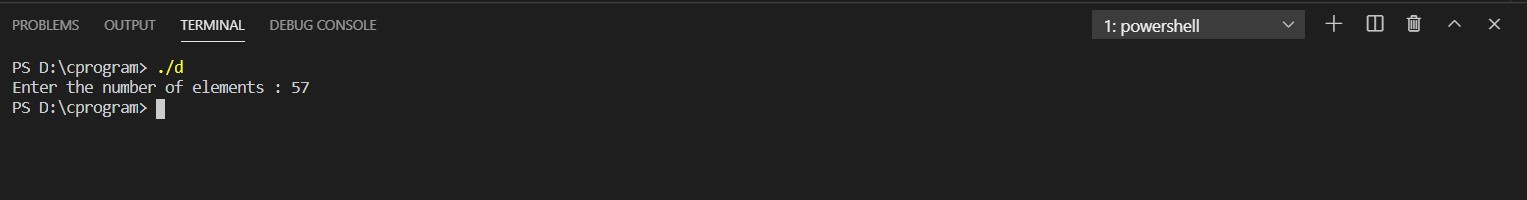
scanf(" %d",&n);

int arr[n]; //variable length array.

}

This code also runs only on GCC.



****

1. Being a newer compiler, GCC merges the literals by default while Turbo C does not.

#include <stdio.h>

int main()

{

    char \*p = "I am Darshak";

    char \*q = "I am Darshak";

    if(p==q)

    {

        printf("Optimized code as GNU based compiler");

    }

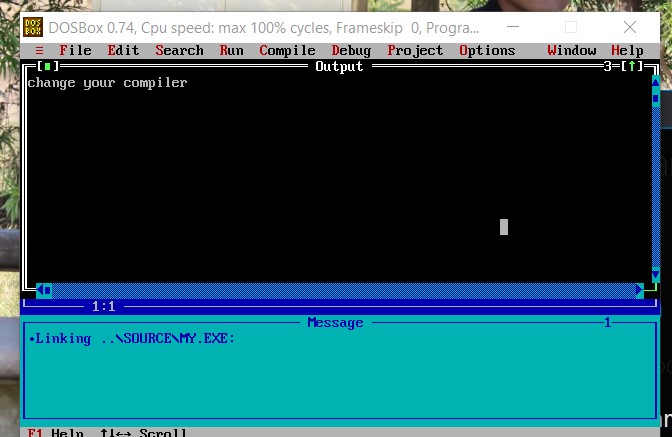
    else{

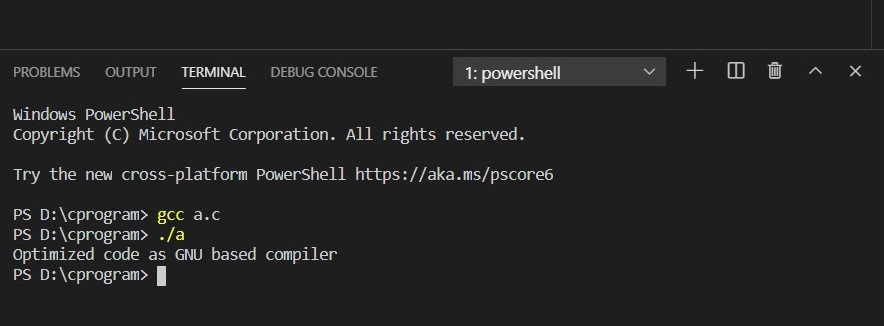
        printf("Change your compiler");

    }

    return 0;

}





It is unspeciﬁed whether these arrays are distinct provided their elements have the appropriate values.

That means it is unspecified whether p and q are pointing to the same string literal or not. In case of GCC they both are pointing to "I am Darshak" (GCC optimizes your code) whereas in turbo c they are not.

Since your string literal is a constant expression, i.e. you should not modify it via a pointer, there is no real purpose in storing it in the memory space twice. Being a newer compiler, GCC merges the literals by default while Turbo C does not.

Both compiler support merging string constants as an option. The GCC option (-fmerge-constants) is turned on at optimization levels, while the Turbo C Option (-d) is turned off on default. If you are using the TCC IDE, then go to Options|Compiler...|Code Generation.. and check "Duplicate strings merged".

GCC is fine but try below program  
o/p will different in Turbo C and GC

#include<stdio.h>

int main()

{

    int a=1,b=1;

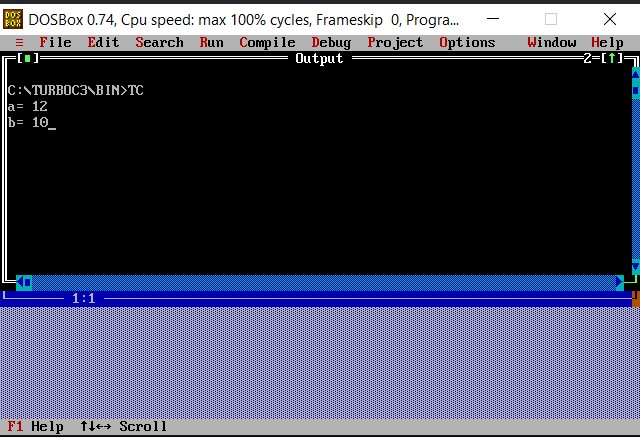
    a=++a + ++a + ++a;

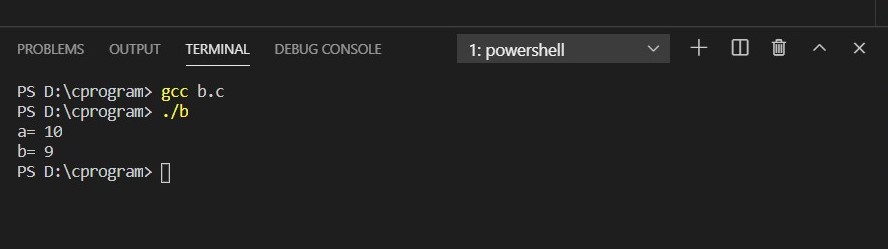
    b=++b + ++b + b++;

    printf("a= %d\nb= %d",a,b);

    return 0;

}





There is a concrete explanation in case of **Turbo C** compiler for this case.

While **GCC** or other modern day compilers have an **undefined behavior** for such expressions where such ambiguity persists.

SO, here this problem occured. Output is different. In Turbo C and GCC based compiler.