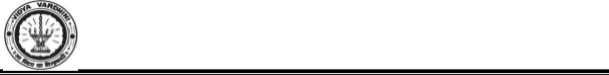


|  |  |
| --- | --- |
| **Name:** | BARI ANKIT VINOD |
| **RollNo:** | 65 |
| **Class/Sem:** | SE/IV |
| **ExperimentNo.:** | 8 |
| **Title:** | Mixedlanguageprogramtoaddtwonumbers |
| **DateofPerformance:** | 06/03/24 |
| **DateofSubmission:** | 22/03/24 |
| **Marks:** |  |
| **SignofFaculty:** |  |



**Aim:**Mixedlanguageprogramforaddingtwonumbers.

**Theory:**

Cgeneratesanobjectcodethatisextremelyfastandcompactbut itisnotasfastas theobjectcode generatedbyagoodprogrammer usingassemblylanguage.Thetimeneededtowriteaprogramin assemblylanguageismuchmorethanthetimetakeninhigherlevellanguageslikeC.

However,therearespecialcaseswherafunctioniscodedinassemblylanguagetoreducetheexecution time.

Eg:Thefloatingpointmathpackagemustbeloadedassemblylanguageasitisusedfrequentlyandits executionspeedwillhavegreateffectontheoverallspeedoftheprogramthatusesit.

Thereare alsosituationsinwhichspecialhardwaredevicesneedexacttiminganditismusttowritea programinassemblylanguagetomeetthisstricttimingrequirement.Certaininstructionscannotbe executedbyaCprogram

Eg:ThereisnobuiltinbitwiserotateoperationinC.Toefficientlyperformthisitisnecessarytouse assemblylanguageroutine.

Inspite ofCbeingverypowerful,routinesmustbewritteninassemblylanguageto:

1.Increasethespeedandefficiencyoftheroutine

2.PerformmachinespecificfunctionnotavailableinMicrosoftCorTurboC.

3.Usethirdpartyroutines

CombiningCandassembly:

Built-In-InlineassemblesisusedtoincludeassemblylanguageroutinesinCprogramwithoutanyneed foraspecificassembler.

Suchassemblylanguageroutinesarecalledin-lineassembly.

TheyarecompiledrightalongwithCroutinesratherthanbeingassembledseparatelyandthenlinked togetherusinglinkermodules providedbytheCcompiler.

TurboChasinlineassembles.

Inmixed languageprogram,prefixthekeywordasmforafunction andwriteAssembly instructionin thecurlybracesin aCprogram

**Code :**

#include<stdio.h>

#include<conio.h>

void main(){

int a, b, c;

clrscr();

printf("\nEnter number a :");

scanf("%d", &a);

printf("\nEnter number b :");

scanf("%d", &b);

asm{

mov ax, a

mov bx, b

sub ax, bx

mov c, ax

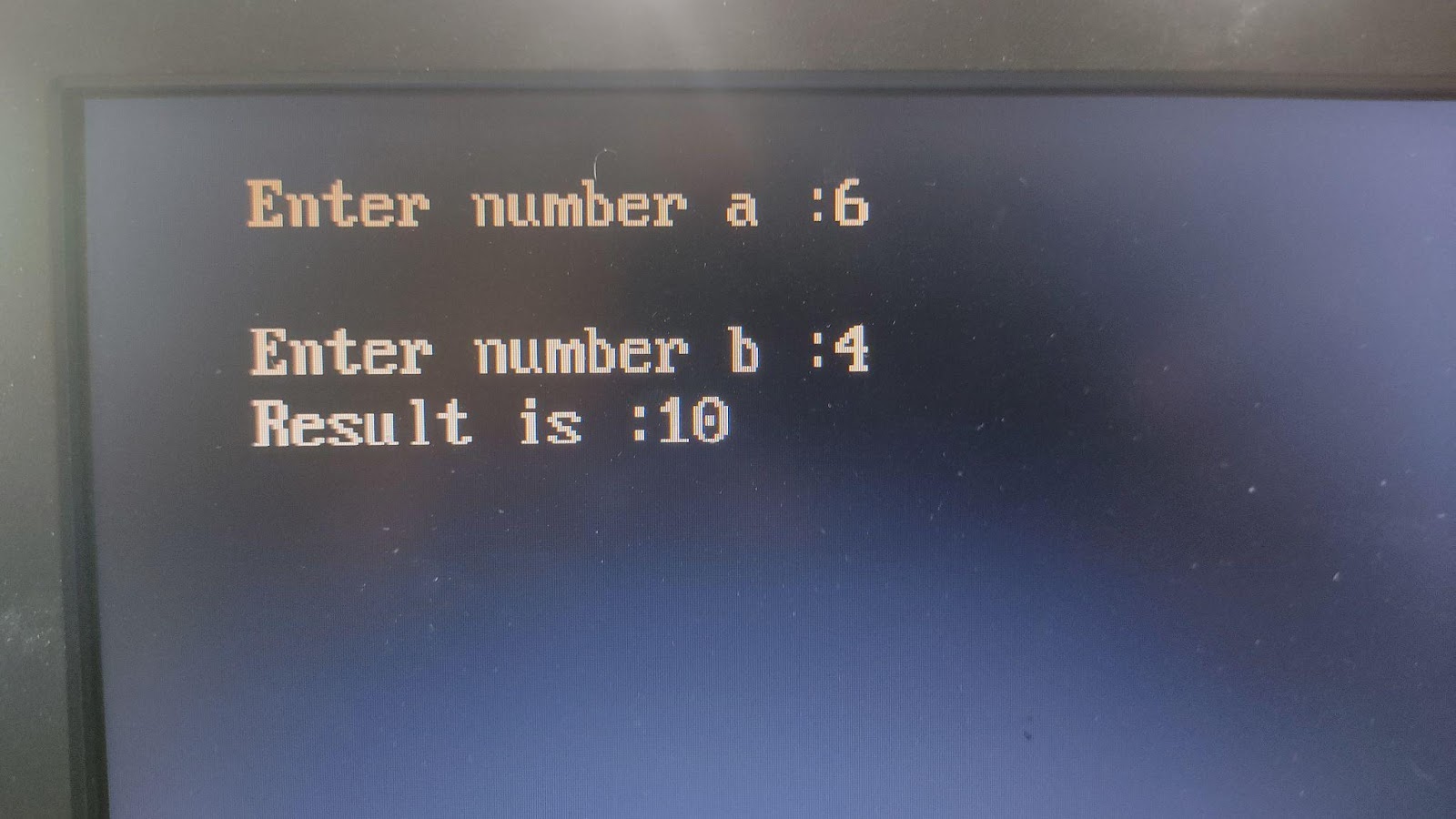
}

printf("Result is :%d", c);

getch();

}

**Output :**



**Conclusion :**

In conclusion, the development of a mixed-language program to add two numbers underscores the versatility and efficiency of leveraging multiple programming languages. By seamlessly integrating the strengths of different languages, such as Python, C, or Java, we can harness their respective capabilities to optimize performance, enhance functionality, and streamline development processes. Through this project, we have demonstrated the power of collaboration across language boundaries, paving the way for future innovations and solutions in software development. As technology continues to evolve, the synergy between diverse programming languages will remain instrumental in addressing complex challenges and driving progress in the digital landscape.