برنامه‌سازی پیشرفته در محیط

Visual C++

مولف:

دکتر محمد علی ترکمانی

مهندس الهام ادراکی

سرشناسه	:	ترکمانی، محمدعلی، ۱۳۵۴ -
عنوان و نام پدیدآور	:	برنامه‌سازی پیشرفته در محیط ++/ مولفان محمدعلی ترکمانی، الهام ادارکی.Visual C
مشخصات نشر	:	مشهد: ارسطو، ۱۳۹۵.
مشخصات ظاهری	:	۲۴۹ص.: مصور، جدول، نمودار.
شابک	:	978-600-432-080-1
وضعیت فهرست نویسی	:	فیپا
موضوع	:	ویژوال سی++ مایکروسافت
موضوع	:	Microsoft visual C ++
موضوع	:	سی ++ (زبان برنامه‌نویسی کامپیوتر)
موضوع	:	C++ (Computer program language)
شناسه افزوده	:	ادراکی، الهام، ۱۳۵۶ -
رده بندی کنگره	:	۱۳۹۵ ۴ت۹۳س/ QA۷۶/۷۳
رده بندی دیویی	:	۰۰۵/۲۶
شماره کتابشناسی ملی	:	۴۳۵۹۱۵۳

نام کتاب : برنامه‌سازی پیشرفته در محیط Visual C++

موضوع: برنامه نویسی Visual C++

موضوع: Visual Studio.net

مؤلفان : دکتر محمدعلی ترکمانی –مهندس الهام ادراکی

ناشر : ارسطو (با همکاری سامانه اطلاع‌رسانی چاپ و نشر ایران)

صفحه‌آرایی، تنظیم و طرح جلد: علی بیات

تیراژ : ۱۰۰۰ جلد

نوبت چاپ : چهارم – ۱۳۹۸

تعداد صفحات: ۲۳۴ ص

چاپ : مدیران

قیمت : ۴۵۰۰۰ تومان

تلفن‌های مرکز پخش : ۰۹۱۷۷۱۶۴۹۴۰ – ۵۰۹۶۱۴۶ - ۰۵۱۱

فهرست مطالب

فصل اول:مروری بر برنامه‌سازی ساخت‌یافته در C++..... ۱۱

فصل چهارم:برنامه‌سازی شیءگرا..... ۸۷

مقدمه:

برنامه‌سازی پیشرفته و برنامه‌سازی شیءگرا دروسی هستند که در رشته‌های مهندسی کامپیوتر، IT و ICT ارائه می‌گردند. با توجه به نیاز دانشجویان و اساتید به مرجعی مناسب این کتاب ارائه گردیدو این کتاب برخلاف سایر کتابهای موجود در بازار از ویژگیهای زیر برخوردار است:

۱-پوشش سرفصلهای مصوب وزارت علوم

۲-حجم مناسب و قابل ارائه در یک نیمسال تحصیلی

۳-بیان ساده و کاربردی

۴-به روز بودن مطالب: برنامه های ارائه شده دراین کتاب با آخرین نسخه Visual studio.net سازگار هستند و به راحتی در کنسولVisual C++.NET اجرا میشوند.

۵-رائه نمونه سئوال در پایان هر بخش که به دانشجویان کمک میکند بهتر بتوانند خود را برای آزمون‌هایی که در پیش رو دارند آماده نمایند.

امید است این اثر مورد توجه همکاران و دانشجویان گرامی قرار گرفته و به افزایش کیفیت آموزشی دروس برنامه‌سازی پیشرفته و برنامه‌سازی شیءگرا کمک نماید. از اساتید و دانشجویان گرامی تقاضا داریم نقطه نظرات خود را از طریق ایمیل m.a.torkamani@gmail.com با مولفین در میان بگذارند تا انشالله در ویرایش‌های بعدی اشکالات یا کاستی‌های احتمالی کتاب مورد تجدید نظر قرار گیرد. در پایان وظیفه خود می‌دانم از زحمات آقای مهندس بیات به خاطر طراحی جلد کتاب و همچنین از مدیریت سامانه اطلاع رسانی چاپ و نشر ایران جناب آقای حسین قنبری تشکر و قدردانی نمایم.

محمد علی ترکمانی

تابستان۱۳۹۵

فصل اول

مروری بر برنامه‌سازی ساخت‌یافته در C++

۱-۱-زبان C++

C++ گونه توسعه یافته زبان C است که در اوایل دهه ۱۹۸۰ توسط بیارنه استراوسروپ[1] در آزمایشگاه بل ایجاد گردید. C++ شامل تمامی امکانات زبان C است وعلاوه بر آن دارای امکانات برنامه‌نویسی شیءگرا(OOP)[2] می‌باشد(شکل ۱-۱). بنابراین می‌توان یک برنامه C را با کامپایلر C++ کامپایل کرد .

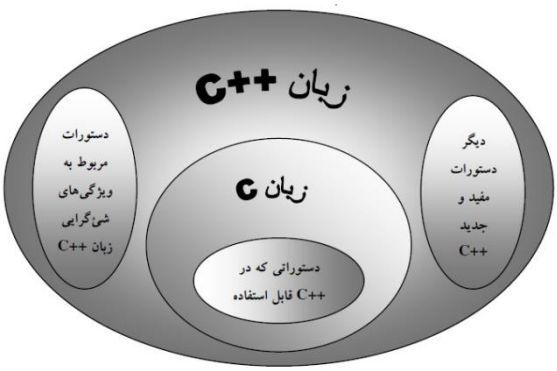
تاکنون کامپایلرهای مختلفی از C++ به بازار عرضه شده است که مهم‌ترین آنها عبارتند از Turbo C++ 3 محصول شرکت Boarland، Visual C++ 6 و Visual C++.Net که محصول شرکت ماکروسافت می‌باشند. Visual C++.Net جزیی از بسته نرم‌افزاری Visual Studio.Net است. در این مجموعه ابزارهای قدرتمند برنامه نویسی زیر وجود دارد:

- Visual C++.Net
- Visual C#.Net
- Visual F# .Net
- Visual Basic .Net

در این کتاب از کامپایلر Visual C++.Net 2010 استفاده شده است.

---

1 Bjarne Srroustrup
2 object oriented programming



شکل ۱-۱: رابطه برنامه‌نویسی C و C++

۲-۱-متغیرها

در زبان C++ هر متغیر، پیش از آنکه در دستوری از برنامه به کار برده شود، باید تعریف گردد. انواع متغیرها در زبان C++ در جدول ۲-۲ نمایش داده شده است.

جدول۱- ۱: انواع متغیرها در زبان C++

نوع	بایت	محدوده مقادیر
int	4	2,147,483,647تا–2,147,483,648
unsigned int/ unsigned unsigned long /unsigned long int	4	4,294,967,295تا0
bool	1	trueیاfalse
char	1	127تا–128
unsigned char	1	255تا0
short/short int	2	32,767تا–32,768
unsigned short/ unsigned short int	2	65,535تا0
long/ long int	4	2,147,483,647تا–2,147,483,648
long long	8	تا–9,223,372,036,854,775,808 9,223,372,036,854,775,807 معادل با یک عدد صحیح ۶۴ بیتی است
float	4	3.4E +/- 38 (7رقم)
double / long double	8	1.7E +/- 308 (15رقم)
wchar_t	2	دو بایت برای ذخیره کاراکترهای Unicode

۳-۱-ساختار برنامه C++

هر برنامه C++ دارای یک یا چند تابع است که یکی از آنها برنامه اصلی یا main است. درC++ همه دستورات و زیر برنامه‌ها به عنوان یک تابع می‌باشند. شروع برنامه از تابع main می‌باشد. یک بلاک در C++ با آکولاد {} مشخص می‌شود.

به طور کلی می‌توان گفت که هر تابع C++ حداقل شامل اجزای زیر است:

```
int main(int argc, _TCHAR* argv[])
{ شروع تابع اصلی
variables declaration; // تعریف متغیرها
program statements;    //دستورات برنامه
return 0;
} پایان تابع اصلی
```

تعریف توابع دیگر ممکن است قبل یا بعد از تابع اصلی قرار گیرد که در قسمت توابع آنرا شرح می‌دهیم.

درون پرانتز آرگومان‌های ورودی است و قبل از main نیز نوع بازگشتی را نشان می‌دهد.

۴-۱-توضیحات[1] در زبان C++

در زبان C++ توضیحات را باید بعد از // قرار داد.

مثال:

```
//program by M.A.Torkamani
```

نکته: در زبان C توضیحات را باید در بین / * و * / قرار دهیم. در زبان C++ نیز استفاده از این روش قابل قبول است.

```
/* sum of two numbers */
```

[1] Comment

۵-۱-دستورات ورودی- خروجی

۱-۵-۱-دستور printf

این فرمان جهت چاپ اطلاعات روی صفحه نمایش به کار می‌رود.

شکل کلی دستور به صورت زیر است:

int printf(const char *control_string, ...);

این تابع یک عدد را بر می‌گرداند که این عدد تعداد کاراکترهای چاپ شده روی صفحه است و یا یک عدد منفی است که نشان دهنده رخ دادن خطا است.

control_string می‌تواند دو نوع آیتم باشد. نوع اول رشته‌ای است که قرار است چاپ شود و نوع دوم تعیین کننده فرمت[۱] است که چگونگی نمایش آرگومان روی صفحه را مشخص می‌کند و با یک علامت درصد شروع می‌شود. ابتدا مثالی از نوع اول ارائه می‌گردد. تعیین کننده فرمت بعدا شرح داده می‌شود.

مثال: برنامه زیر موجب چاپ رشته hello روی مانیتور خواهد شد.

```
#include "stdafx.h"
int main(int argc, _TCHAR* argv[])
{
  printf ("hello") ;
return 0;
}
```

نکته: یک هدر فایل[۲] فایلی با پسوند h. است که یک سری توابع در آن تعریف شده است و ما می‌توانیم از آن استفاده کنیم. هدر فایل‌ها به وسیله فرمان #include به برنامه ما ضمیمه می‌شوند. هر یک از فرمان‌های زبان C++ درون یک یا چند فایل که با نام header file (یا library) شناخته می‌شود، تعریف شده است. برای اینکه برنامه بدون خطا اجرا شود باید header file فرمان مورد نظر را قبل از تابع main بنویسید. به عنوان مثال دستورات پر کاربرد زبان Visual C++ 2010 درون کتابخانه "stdafx.h" تعریف شده‌اند. برای استفاده از این هدر فایل باید فرمان #include "stdafx.h" را قبل از تابع main بنویسید.

۱Format Specifier

۲header file

تذکر: وقتی که برنامه به اتمام می‌رسد به محیط C++ بر می‌گردد. جهت دیدن جواب‌ها و برگشتن به C++، یعنی رفت و آمد بین محیط C++ و کنسول (محیط مشکی رنگ شبیه به محیط DOS) از CTRL+F5 استفاده می‌شود.

تذکر: اگر بخواهید وقتی که برنامه به پایان رسید، در محیط C++ باقی بماند و وقتی که یک کلید را فشار دهید به محیط C++ برگردد، باید دستور getch_ را در انتهای برنامه بنویسید. این دستور منتظر می‌ماند تا شما یک کلید را فشار دهید. این دستور در هدر فایل conio.h تعریف شده است:

```
#include "stdafx.h"
#include "conio.h"
int main(int argc, _TCHAR* argv[])
{
  printf ("hello") ;
  _getch();
return 0;
}
```

۱-۱-۵-۱-کنترل عمل چاپ در دستور printf

جدول ۱-۲ برخی از کاراکترهای کنترلی مهم که در دستور printf قابل استفاده هستند را نشان می‌دهد.

جدول ۱- ۲: کاراکترهای کنترلی در زبان C++

ردیف	علامت	توضیح
۱	\n (new line)	به ابتدای خط بعدی می‌رود .
۲	\t (tab)	8 ستون به جلو می‌رود.
۳	\"	برا ی چاپ " به کار می‌رود .
۴	\\	برا ی چاپ \ به کار می‌رود.
۵	\b (back space)	یک حرف را پاک می‌کند.

مثال: برنامه زیر را اجرا کرده و خروجی آنرا مشاهده نمایید.

```
#include "stdafx.h"
#include "conio.h"
int main(int argc, _TCHAR* argv[])
{
  int a;
```

```
  printf("salam\b ali");
  printf("\nsalam\\ ali");
  printf("\nsalam\t ali");
  printf("\nsalam\" ali");
  printf("\nsalam\n ali");
  _getch();
  return 0;
 }
```

۲-۱-۵-۱-تعیین کننده‌های فرمت

فرض کنید بخواهید مقدار متغیر A را چاپ کنید. اگر فرمان زیر را به کار ببرید، کامپیوتر حرف A را روی صفحه چاپ می‌کند:

```
  printf("A");
```

برای اینکه به کامپیوتر بفهمانیم که می‌خواهیم مقدار متغیر A را چاپ کنیم و نه حرف A را از یک تعیین کننده فرمت استفاده کنیم. تعیین کننده‌های فرمت در جدول ۱-۳ ارائه شده‌اند.

بنابراین اگر بخواهیم یک حرف را چاپ کنیم، آنرا درکتیشن دوتایی قرار می‌دهیم، اما اگر بخواهیم یک متغیر را چاپ کنیم باید از درصد متناسب با نوع آن متغیر استفاده کنیم.

مثال:

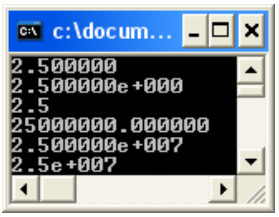
```
int A ;
A=2;
printf (“A” ) ;          // حرف A را چا پ می کند.
printf ( “%d”  , A);      //2 را چا پ می کند.
```

مثال: اگر B=2.5*10000000 و A=2.5 باشد به کار بردن %f ، %e و %g موجب چاپ چه چیزی می‌شود؟

```
#include "stdafx.h"
#include "conio.h"
int main(int argc, _TCHAR* argv[])
{
  float a=2.5;
  printf("%f", a);
  printf("\n%e", a);
  printf("\n%g", a);
  float b=2.5*10000000;
  printf("\n%f", b);
  printf("\n%e", b);
  printf("\n%g", b);
  _getch();
```

```
return 0;
}
```

خروجی برنامه:



جدول ۱- ۳: تعیین کننده‌های فرمت

فرمت خروجی	نوع	کاراکتر
یک کاراکتر	int	c
یک عدد صحیح	int	d
مبنای ۸	int	o
اعداد صحیح بدون علامت	int	u
مبنای ۱۶ با استفاده از حروف abcdef	int	x
مبنای ۱۶ با استفاده از حروف ABCDEF	int	X
نماد علمی (توانی)	double	e
double یا float (معمولی)	double, float	f
توانی یا معمو لی، هر کدام که ساده تر باشد نوشته می‌شود	double	g
آدرس آرگومان را به صورت مبنای ۱۶ چاپ می‌کند.	Pointer to void	p
رشته‌ها	string	s

۲-۵-۱-دستور scanf

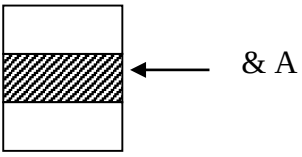
این دستور برای دریافت اطلاعات از ورودی به کار می‌رود. شکل کلی این دستور به صورت زیر است:

int scanf (const char *control_string, ...);

تابع scanf یک عدد را بر می‌گرداند که این عدد تعداد آیتم‌هایی است که با موفقیت به یک متغیر نسبت داده شده‌اند. در اینجا control_string نحوه خواندن متغیرها از لیست

آرگومانهای تابع را نشان می‌دهد. برای control_string می‌توان از همان درصدهای ذکر شده در جدول ۲-۴ استفاده نمود.

نکته مهم در مورد scanf این است که شکل کلی این دستور شبیه به printf است، با این تفاوت که در این دستور به جای کار کردن با خود متغیر، با خانه‌ای از حافظه که متغیر در آن قرار دارد، کار می‌کنیم. برای انجام این کار، از کاراکتر & (امپرسن) استفاده می‌کنیم. اگر متغیری به نام A داشته باشیم، &A به آدرس A در حافظه اشاره می‌کند. &A در واقع یک اشاره‌گر[۱] به خانه‌ای از حافظه است که A در آن قراردارد (شکل ۱-۲).



شکل ۱- ۲: اشاره‌گر به آدرس متغیر A

مثال:

```
int   A;
scanf("% d", &A) ;
```

نکته: وارد کردن داده‌ها دقیقا باید مانند آنچه در دستور scanf_s_s است، انجام شود.

مثال:

```
scanf("%f ,%f", &B,&A);
```

در اینجا باید داده‌ها را به صورت زیر واردکنیم:

```
5 , 7
```

در B مقدار ۷ و در A مقدار ۵ به فرمت اعشاری ذخیره می‌گردد.

۶-۱-عملگرها

در زبان visual c++ عملگرها به چند دسته تقسیم می‌شوند:

- عملگرهای محاسباتی

۱pointer

- عملگرهای رابطه‌ای
- عملگرهای منطقی
- عملگرهای بیتی
- عملگرهای ترکیبی

۱-۶-۱- عملگرهای محاسباتی

عملگرهای محاسباتی در جدول۱-۴ نمایش داده شده‌اند.

جدول ۱- ۴: عملگرهای محاسباتی

عملگر	وظیفه عملگر	مثال
-	تفریق یا منهای یکانی	x-y یا –x
+	جمع	x+y
*	ضرب	X*y
/	خارج قسمت تقسیم	x/y
%	باقیمانده تقسیم صحیح	x%y
--	کاهش یک واحد	--x یا x--
++	افزایش یک واحد	++x یا x++

۲-۶-۱-عملگرهای رابطه‌ای

این نوع عملگرها رابطه بین عملوندها را مشخص می‌کنند. جدول ۱-۵ عملگرهای رابطه‌ای را نشان می‌دهد.

جدول ۱- ۵: عملگرهای رابطه‌ای

عملگر	رابطه
>	بزرگتر
> =	بزرگتر یا مساوی
<	کوچکتر
< =	کوچکتر یا مساوی
==	مساوی
! =	نامساوی (مخالف با)

تذکر: = برای جایگزینی و == برای مقایسه (در دستورات شرطی) به کار می‌رود.

مثال: عبارت منطقی آیا x با y برابر است، به صورت x==y نوشته می‌شود.

تذکر: در زبان C++ نتیجه یک عبارت شرطی درست ۱ و یک عبارت شرطی غلط صفر است.

۳-۶-۱-عملگرهای منطقی

عبارات منطقی عباراتی هستند که دارای ارزش درست یا غلط است. جدول ۱-۶ عملگرهای منطقی به ترتیب اولویت را نمایش می‌دهد.

جدول ۱- ۶: عملگرهای منطقی

عملگر	نام	مثال
!	نقیض (not)	!X
&&	و (and)	x>y && m<n
||	یا (or)	m<n||x>y

۴-۶-۱-عملگرهای بیتی[1]

عملگرهای بیتی، عملگرهایی هستند که برای کار روی بیتها استفاده می‌شوند. جدول ۱-۷ اولویت عملگرهای بیتی را نشان می‌دهد.

جدول ۱- ۷: اولویت عملگرهای بیتی

عملگر	وظیفه
&	And
|	OR
^	XOR
~	Not
>>	یا شیفت به راست Right Shift
<<	یا شیفت به چپ Left Shift

مثال: شیفت به راست و چپ

```
#include "stdafx.h"
#include "conio.h"
```

1 Bitwise operators

```
int main(int argc, _TCHAR* argv[])
{
  unsigned char C=4;
  C=C >> 1;
  printf("\n%d",C);
  C=8;
  C=C << 1;
  printf("\n%d",C);
  _getch();
  return 0;
 }
```

خروجی برنامه فوق به ترتیب ۲ و ۱۶ است.

۵-۶-۱-عملگرهای ترکیبی

این عملگرها از ترکیب عملگرهای محاسباتی و علامت = ایجاد می‌شوند. کار این عملگرها یک عمل محاسباتی و یک عمل انتساب است. اولویت این عملگرها از سایر عملگرها کمتر است. جدول ۱-۸ این عملگرها را نشان می‌دهد.

جدول ۱- ۸: عملگرهای ترکیبی

دستور	دستور معادل
A + = B ;	A = A + B ;
A - = B ;	A = A - B ;
A *= B ;	A = A * B ;
A / = B ;	A = A/ B ;
A% = B;	A = A %B

۶-۶-۱-تقدم عملگرها

جدول ۱-۹ تقدم عملگرها را نشان می‌دهد. در این جدول،عملگرهایی که در یک ردیف هستند دارای اولویت یکسان هستند. این عملگرها هر کدام که زودتر ظاهر شوند اول اجرا می‌شوند.

جدول ۱- ۹: تقدم عملگرها

بالاترین اولویت	[] ()
	! ~ ++ --
	* / %

	- +
	<< >>
	<= > >= <
	== !=
	&
	^
	|
	&&
	||
پایین ترین اولویت	= += -= *= /= %=

۷-۱-توابع ریاضی

توابع ریاضی نظیر قدر مطلق (abs)، جذر یا ریشه دوم (sqrt) و توان‌رسانی(pow)، sin و cos در هدر فایل math.h تعریف شده‌اند.

تمرین: برنامه‌ای بنویسید که دو عدد را از ورودی گرفته و به وسیله تابع pow، عدد اول را به توان عدد دوم برساند.

۸-۱-محاسبه خارج قسمت و باقیمانده تقسیم

همانطور که ذکر شد برای محاسبه خارج قسمت و باقیمانده تقسیم از عملگرهای/ و % استفاده می‌شود. اگر A و B از نوع صحیح باشد، A/B از نوع صحیح و اگر B و A از نوع اعشاری باشد، A/B از نوع اعشاری است.

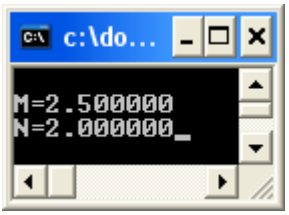
مثال:

```
int  A = 5 ,B = 2 ;
                float M;
M = (float)   A/B;
```

در واقع می‌گوییم که نوع A /B را از نوع float در نظر بگیرد. یعنی عمل تقسیم را به صورت اعشاری انجام بده. اگر (float) را ننویسیم به جای 2.5، عدد صحیح 2 را بر می‌گرداند. به این کار Type Casting می‌گویند. به برنامه زیر توجه کنید:

```
#include "stdafx.h"
#include "conio.h"
int main(int argc, _TCHAR* argv[])
{
   int  A = 5 ,B = 2 ;
   float M,N;
    N =    A/B;
       M = (float)   A/B;
  printf("\nM=%f\nN=%f",M,N);
  _getch();
  return 0;
}
```

خروجی برنامه:



۹-۱-مروری بر دستورات مهم زبان C++

۱-۹-۱- پاک کردن صفحه کنسول

در Viusal C++ .NET برای پاک کردن صفحه کنسول از دستور زیر استفاده می‌شود.

system("cls");

توجه داشته باشید که برای استفاده از این دستور باید کتابخانه‌ای که این دستور در آن تعریف شده است را در بالای برنامه تعریف کنید:

#include <stdlib.h>

۲-۹-۱-دستور for

این دستور برای ایجاد حلقه به کار می‌رود. به عبارت دیگر وقتی از این دستور استفاده می‌شود که بخواهیم یک یا چند دستور به بیش از یک مرتبه(به عنوان مثال ۱۰۰ مرتبه) اجرا شوند.

حلقه for دارای یک شاخص حلقه است که شرط خروج از حلقه بر اساس آن شاخص تعیین می‌شود.

شکل کلی دستور به صورت زیر است:

for(*initialization*; *condition*; *increment*) *statement*;

که در آن:

initialization: مقدار اولیه شاخص حلقه

condition: شرط پایان حلقه

increment: گام حلقه که میزان افزایش یا کاهش شاخص حلقه را مشخص می‌کند.

مثال:

```
for (i=5 ;  i< =100; i + = 7)

بدنه حلقه
}
```

در این مثال ۵ مقدار اولیه، ۱۰۰ مقدار نهایی و ۷ گام یا step حلقه است.

مثال: در قطعه برنامه زیر شاخص حلقه فقط یک واحد اضافه شود:

```
for (i=0 ;  i< =10; i ++)

بدنه حلقه
}
```

تذکر: اگر در بدنه for فقط یک دستورالعمل وجود داشته باشد، نیازی به نوشتن آکولاد نیست.

مثال:

```
for (i = 1; i <50;  i++)
    {
        s=s + i;
      }
```

برنامه فوق را می توان به صورت زیر نوشت:

```
for (i = 1; i< 50; i++)
s = s   +  i   ;
```

مثال: برنامه‌ای بنویسید که مجموع اعداد 5- تا 10 را محاسبه و چاپ نمایید.

```
#include "stdafx.h"
#include "conio.h"
int main(int argc, _TCHAR* argv[])
{
int   i , A=0  ;
for  (i  = -5 ;  i <= 10 ; i ++)
A += i;
 printf ("%d", A );

_getch();
return 0;
}
```

۱-۲-۹-۱-حلقه بینهایت با استفا ده از for

```
for (  ;   ; )
{
}
```

به وسیله دستور break می‌توان از حلقه بینهایت خارج شد. بعدا با این دستور آشنا خواهید شد.

نکته: شاخص حلقه for را می‌توان در خود حلقه تعریف کرد. در این صورت متغیر فقط در داخل حلقه قابل استفاده است.

مثال:

```
for (int i=1;i<=10;i++)
```

۲-۲-۹-۱-حلقه for تو در تو[1]

حلقه for می‌تواند به صورت تو در تو به کار رود:

```
int i , j;
   for (i=1;i<=10;i++)
   {
       for (j=1;j<=10;j++)
```

1 Nested For