

جزوه درسی طراحی و پیاده سازی زبان های برنامه سازی استاد عسکری - کارشناسی کامپیوتر

این قابلیت را برای برنامه نویسی زبان فراموش نکنیم بعد از اینها می توانیم

۱- بیاییم زبان را برنام نویسی آن دارای معنا و معنی باشد.
مثال: دستگیر $(a+b) \text{ write } (a \times b)$ در زبان با معنای دارد و معنی می دهد که بخار در این
دستگیر $(a+b) \text{ write } (a \times b)$ و معنای رابطه ای مخالف بودن دو مقدار است و معنی می دهد که با هم
تربیب شدند.

۳- بتوانیم در زبان برنامه نویسی با تعداد مقول که کمتری فوایدی دارد یا با سازگاری خاشیم که این امر
محو لا در زبان های سطح بالا فراموش می شود.


۱2
مثال: فرض کنید می خواهیم در زبان ما سین و اسمبل جزای بنویسیم که ما می بینیم
بیجا که نامناسب در عدد را در عددی دریافت کند و حاصل جمع آن ها را با نامش بیجا نامناسب
دیده در جدول نمایش دهد که تربیب به محافظه گد نیاز خواهیم داشت ولی همین برنامه را فراموش
کنیم می خواهیم در زبان سطح بالا ما $C++$ یا C قرار بنویسیم
۱3
که تعداد گد های آن نسبت به زبان سطح پایین اسمبل مقایله به $\frac{1}{10}$ می باشد یعنی بسیار کند.

۱4
۴- و منبع فراموشی مناسب را در مثال به عنوان دستگیر داریم زبان برنامه نویسی فعلی که
۱5
نظم گرفت.

۱6
مثال: محیط زبان های Borland, Pascal دارد و معنی می دهد که به صورت یک صفحه ای می توان
همراه با یک مکان های نزدیک به تعداد در مقایله محیط فوق زبان و در مثال بیست و یک را در مثال فراموش
گرفت که دارای وجود کمترین مقول تری می باشد که این امر محولا در زبان های سطح بالا
فراموش شده است.

تاریخچه
۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱ ۳۲

تأسیس سپاه پاسداران انقلاب اسلامی ۱۳۵۸ هـ.ش - سازمان نظام انقلاب فرهنگی ۱۳۵۹ هـ.ش
به جان خواجه و حق قدیم و عهد دوست
که مولود است



جزوه درسی طراحی و پیاده سازی زبان های برنامه سازی استاد عسکری - کارشناسی کامپیوتر

۱- قابلیت عمل یا Portable بودن معنای یک زبان

برنامه سازی نیابتی وابسته به ماشین یا کامپیوتری خاص یا فقط در این معنای امر برنامه های عمل سیستم کامپیوتری در یک زبان برنامه سازی نوشته می شود و انتقال به کامپیوتری دیگر انتقال دهیم که این امر برای برنامه نویسان زبان مطلوب نمی باشد.

۲- با این بودن هزینه زبان برنامه سازی؟

ظلم رقم

درمان زبان های برنامه سازی هزینه های قابل از ۳۰ تا ۳۰۰ هزار دلار مورد بررسی قرار داده

- ۱- هزینه ترجمه
- ۲- هزینه اجرا
- ۳- هزینه نگهداری

هزینه ترجمه معمولاً در زبان های سطح می شود که در محیط های کاربردی مورد استفاده قرار می گیرد زیرا برنامه های نوشته شده در این دسته از زبان ها معمولاً دارای فضای باقی مانده برای توسعه بیشتر و امکان با تغییر هم خواهند شد و هزینه اجرای زبان های مهم فضا محدود در طول روز به دفعات زیاد اجرا می شود

نظریه نگهداری: همان طور که می دانید در طول عمر یک محصول نرم افزار این هزینه نگهداری می تواند دارای اهمیت بیش تری باشد زیرا دوره نگهداری محصول نرم افزار در طول عمر آن بیشترین دوره را به خود اختصاص می دهد بنابراین محصول خوب خواهد بود که دوره نگهداری آن دارای کمترین هزینه باشد و نگهداری یک محصول نرم افزار می تواند به ۳ نوع نگهداری تقسیم (perfective)

نگهداری اصلاح فرین (corrective) نگهداری تطبیقی (Adaptive) بخش خایه پس از گذشت زمان کرده تغییر یافته می - محصول نرم افزار می خواهد قابلیت های جدیدی را به یک محصول نرم افزار اعمال کند تا بتواند اعمال جدیدی را در آن انجام دهد و زبان برنامه سازی که امکان

نگهداری تطبیقی گفته می شود

M	T	W	T	F	S	S
	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					



اصلاحی: همان طوره که می دانید یک معضل نرم افزار و قبل از
و آنگاه دارد فضاها

24 Apr. Wed. 2019

۴

۱۸ شعبان ۱۴۴۰

چهارشنبه اردیبهشت

۹۸/۲/۴ هفته ۲۳۰-۲۳۵

تعمیر به دست ما

از معضل در آن مرحله بعدی مانند با امروز همان و کار کردن کار به آن با معضل فضاها تا این
می نشیند تا به این وظیفه مرده توسعه دهنده می معضل می باشد که فضاها می شود نظر را اصلاح نماید
تطبیق: فرض کنید زبان برنامه سازی طراحی و تولید شود و با ارائه یک سیستم عامل
جدید این زبان قابلیت اجرا شدن در آن سیستم عامل را نداشته باشد تا به این
وظیفه توسعه دهنده آن زبان برنامه سازی می باشد که تغییراتی در آن اعمال نماید تا زبان های
برنامه سازی قابلیت تطبیق با معضل جدید را داشته باشند.

دلایل صلاحیت زبان های برنامه سازی:

- ۱- انتخاب سیستم زبان: اجرای نوشتن یک برنامه به عنوان مثال فرض کنید بنویسیم
یک وب سایت طراحی کنیم و با توجه به شناخت زبان های برنامه سازی که با مطالعه آنجا
حاصل می شود در سیستم زبان را برای طراحی آن وب سایت انتخاب کنید.
- ۲- انتخاب سیستم ساختار داده: برای نوشتن برنامه در یک زبان خاص به عنوان مثال
فرض کنید در زبان ++C بنویسیم که یک عدد حقیقی را از ورودی دریافت
کنیم و معادله $x^2 + 4x + 4 = 0$ را حل کنیم و در خروجی آن عدد را نمایش دهیم راهکار پیش یا انتخاب برای نوشتن این برنامه
اعمال ۱۰۰۰۰ دستور ۱۴ خواهد بود که آلفا سیستم زبان برنامه سازی که گفته است و الله اعلم
برای نوشتن این برنامه انتخاب از معضم تابع خواهد بود که در این راهکار برنامه فوق را می توان
با ۷۰ الی ۷۰۰ خطا کدباز سازی نمود

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰
۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰



مدل‌های زبان های برنامه سازی

۱- مدل دستوری: این مدل زبان های برنامه سازی ترکیبی

از دستورات مختلف می باشد و این دستورات به ترتیب از دستور اول تا دستور n اجرا می شوند

۲- مدل فلهی: در این مدل از زبان های برنامه سازی به صورت زیر می توانیم تفهیم

```

instruction 1;
instruction 2;
...
instruction n;
    
```

۲- مدل تابعی: در این مدل از زبان های برنامه سازی درون یک برنامه می توانیم توابع

مختلف تعریف کنیم و نوشتن یک برنامه با مفهوم تابع می تواند دارای
 فرایندی مختلفی باشد مانند C++ و پاسکال Pascal به صورت فلهی در برنامه درون

شکست حمله نظاس آمریکا به ایران در طیس (۱۳۵۱ هـ.ش)

این مدل از زبان های برنامه سازی می تواند به شکل زیر مطرح شود

function n (..... (function 2 (function 1 (Data)))

و مشابه با دستور فوق می توانیم فرم خاصیم که مدل یک برنامه

n عدد تابع برنامه می باشد function 1 تا function n درون

یک برنامه می باشد و n عدد تابع فوق به صورت مداوم از function n تا function 1

درون یک برنامه می باشد و فرادانی می شود که در بالا ترین سطح function 1 یا Data

فرادانی خواهد شد. و فرادانی این تابع به دنبال بار اجرای function 2 خواهد

می شد و این روند تا function n تکرار می شود



M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

۳- مدل هوش مصنوعی: در این مدل از انواع برنامه سازی

27 Apr. Sat. 2019
۲۱ شعبان ۱۴۴۰
شبه اردیبهشت
۹۸/۰۲/۰۷ هفته ۲۲۷-۵۲۸

۸۸ یک برنامه می تواند تعالی منطقی از دستورات شرطی در نظر گرفته شود
۸۹ ما در زبان C و در زبان های matlab و Python نیز قواعد منطقی است یاد سازی

۹۰ الگوریتم های هوش مصنوعی برای برنامه نویسی آسان فراهم شده است و توانیم سرعت
۹۱ تلاشی یک برنامه در مدل هوش مصنوعی را به شکل زیر در نظر بگیریم.

if conditional then action 1

11 if conditional then action 2

۱
۱۲
۱۳

if conditional n then action n

۱۴ ۴- مدل شیء تراشه در این مدل زبان های برنامه سازی می توانیم اشیاء و منطقی تعریف کنیم
۱۵ و سپس قواعد است که در این اشیاء معرفی کنیم ما در زبان های Python و C# می توانیم
تجزیه عمل اول

۱۶ مثال: دستوراتی در زبان ++C بنویسید که همین آن ها یک واحد متغیر x افزایش دهد.

۱۷ (۱) $x = x + 1$

(۲) $x++$

(۳) $++x$

۱۸ (۴) $x += 1$

۱۹ سوال: در یک زبان تابعی می توانیم از دستوراتی که در صورت تابعی
۲۰ داریم تابعی بنویسیم.

→	←	→	←	→	←	→	←	→	←
۶	۵	۴	۳	۲	۱	۸	۷	۹	۱۰
۱۳	۱۲	۱۱	۱۰	۹	۸	۱۵	۱۴	۱۳	۱۲
۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۲۲	۲۱	۲۰	۱۹



دو ترمینال وصل و نقل

خدا چو صورت اسروری دلگشا...

دسته هفتم

نقد یا Intax 8 صورت فلهری دستورات که بر اساس آن خود

گردد که برای عمل با عمل در زبان های مختلف ممکن است تفاوتی داشته باشد

توجه: مثال برین فایده خواهد بود که برای و اعمدگی به نام α از نوع عددی صحیح در جدول

برنامه سازی C++ و Pascal توجه فایده که دارای تفاوتی بر صورت زیر می باشد:

```
C++: int a[10];
```

```
Pascal: a=array [1..10] of integer;
```

مانند به صورت حقوق می توانیم بگوییم زبان پاسکال نسبت به C++ دارای قابلیت

خواندن در C++ نسبت به پاسکال دارای قابلیت نوشتن می باشد.

برای مصفوم خود می توانیم معیارهایی به صورت زیر مطرح کرد 18 - قابلیت خواندن و نوشتن

2 - عدم وجود انواع 8 بدون معیار صورت فلهری دستورات برنامه با سبکی به گونه ای باشد

که از دید برنامه نویس و کامپایلر زبان با معیار ایجاب می شود به مثال زبان Fortran

15 دارای ایجاباتی می باشد که می از ایجابات آن به لیست جدول خصوصاً انواع تابع و اشاره

به فایده ای از آن که چون آن می باشد که در زبان پس از آن (Pascal) ایجاب می شود در نظر

بر طبقه بندی جدید بدون صورت که برای اشاره به فایده ای از آن که در [] اتفاق شده است

17 - سهولت بازیابی 8 صورت فلهری دستورات که زبان می تواند سهولت بازیابی را

18 برای برنامه نویسان فراهم نماید و این سهولت زمانی فراهم می شود که زبان دارای قابلیت

خواندن باشد 4 - سهولت ترجمه 8 در زبان های برنامه سازی هر چه قابلیت نوشتن بیشتر باشد

سهولت ترجمه برای فاز تکلیف تطبیق لغوی یا (Lexical Analyser) تطبیق لغوی کامپایلر فراهم

می نماید زیرا تطبیق لغوی کامپایلر زبان کمتری را میسر می نماید



	M	T	W	T	F	S	S
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30						

خار (token) سپری گفت

29 Apr. Mon. 2019

9

۲۳ شعبان ۱۴۴۰

دوشنبه اردیبهشت ۹۸/۰۲/۰۹ هفته ۱۷-۴۰-۲۳۵

عناصر فزوی زبان

کلمات کلیدی زبان مانند `int` ، `include` ، `return` ، `درزبان` ++C با شماره ها

مانند اسامی متغیرها و ثابت های `double` ، `برنامه` ، `مخاطب` های زبان ، `اعداد` ، `علائق` و غیره

مانند ... و در ... فضای قابل `تعمیر` یا `توضیحات` می تواند عناصر فزوی `زبان`

منفرد هستند

نکته: عناصر اصلی سیستم های `تایپ` `زبان` های `برنامه سازی`

۱- داده های اولیه: هر `زبان` `برنامه سازی` برای `برنامه نویسی` داده های اولیه ای

را فراهم می نماید که `برنامه نویسی` برای `برنامه های` خود می تواند `اشیاء` داده ای `موازین` داده های

اولیه `تولید` نماید مانند `char` - `double` - `int` - `float` - `C++` که `اشیاء` داده اولیه ای

۲- `Void` - `string` - `bool` فراهم نماید `زبان` `برنامه نویسی` `Pascal` که `اشیاء` داده ای

`Boolean` - `Byte` - `string` - `char` - `real` - `integer`

۳- `ترین` در `زبان` `برنامه سازی` `تعمیر` `اشیاء` داده اولیه ای `آن` را `تایپ` می

طبقه بندی

۲- `اعمال اولیه` `زبان` های `برنامه سازی` دارای `تعمیر` `اعمال اولیه ای` می باشند

`برنامه نویسی` `زبان` `تعمیر` `این` `اعمال اولیه` `برای` `نویسن` `برنامه` خود `استفاده` می نماید

`زبان` `برنامه سازی` `تعمیر` `که` `دارای` `تعمیر` `اعمال اولیه` `معناسازی`

عمل	نماد
جمع	+
تفریق	-
ضرب	*
تقسیم	/
درصد	%
افزایش	++



رواق منظر چشم من آشیانه توست

کرم نما و فرود آ که خانه خانه توست

CMP = compiler

JZ = jump zero

JNZ = jump not zero

ZF = zero flag

10

30 Apr, Tue, 2019

۲۴ شعبان ۱۴۴۰

ارديبهشت سه شنبه

هفته ۱۷ | ۳۴۴-۳۲۴ ه ۹۸/۰۲/۸۰

اعمال منطقی با عمل محاسبه نیست فونکشنی از خانه بر داده می باشد

به عنوان مثال فرض کنید می خواهیم در زبان C مقایسه و عملیات بر این

متغیرات در زبان C Ax و Bx را بر روی می خوانیم محاسبه این عمل از دستور `CMP Ax Bx`

استفاده می کنیم که اگر دستور این در زبان ما هم برابر باشد حاصل عمل تفریق فوق صفر

خواهد شد و با دستور شدن این عمل صافی `ZF=1` می شود و در دستورات `JZ` و `JNZ` می توانیم

فرایندی

عملت اول برنامه استفاده می کنیم

۱۳ کنترل تر سبب: زبان های برنامه سازی دارای محاسبه کنترل تر سبب دستورات

برنامه نویسی با تعدد که این نوع از دستورات فرایند دارای انواع مختلفی خواهد بود به عنوان مثال

زبان C++ را در نظر بگیرید که در این زبان می توانیم بر اساسی بنویسیم که عددی را

از ورودی دریافت کرده و قدر مطلق آن در خروجی نمایش دهد

`int x;`

`cin >> x;`

`if (x <= 0) cout << x;`

`else`

`cout << (-x);`

۱۵ بازگردد بر برنامه فوق به کمک دستور `if` همراه `else` عمل کنترل تر سبب صورت

۱۶ گرفته است و خروجی داریم از دستور کنترل تر سبب دستور سوئیچ در این زبان می باشد

که به کمک این دستور می توانیم از `if` های متناظر و تو دلتو بهره ببریم

۱۷ ملاحظه کنید دستبندی برداره تمامی اشیاء دایره مطرح شده درون زبان های برنامه سازی

درون حافظه جای می گیرند که ملاحظه کنید دستبندی برداره درون حافظه که در آن هست به عددی

در فضای ۱۶ خواهد بود و باید بدانند `x` شروع می شود به عنوان مثال برنامه نویسی را در نظر بگیرید

که در این سبب داده `x` درون حافظه می نشیند



روز ملی حقوق زنان

مرا فشان دل از ره تو را چه افتادست

برو به کار خود ای واعظ این چه فریادست

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

```

#include <iostream>
using namespace std;
int main() {
    int x, *p;
    08 cout << endl; // " " << "enter number = ";
    cin >> x;
    p = &x;
    09 cout << endl; // " "
    cout << "it" << " " << x << " ";
    cout << endl; // " " << "it" << "the address of x is = " << p;
    10 return 0;
}

```

۵- **مفهوم عملیاتی** : مفهومی که در اصل شامل تمام متغیرهای برنامه است که می‌توانیم به سبب دسترسی به آن‌ها از زبان برنامه‌نویسی به آن‌ها دسترسی داشته باشیم. این متغیرها می‌توانند با زبان‌های خارجی خود ارتباط برقرار کنند مانند متغیرهای **input** و **output**.

۶- **مدیریت حافظه** : حافظه‌ای که در سیستم اصلی درون سیستم‌های مدیریت مطرح می‌باشد. در این است که مولفه‌های CPU تا حد امکان منقول باشند و به یکدیگر اختصاص داده شوند. در هر قسمتی از این اصل مطرح می‌شود که در آن سرعت پردازش CPU در مقایسه با دستمای برنامه‌ها درون حافظه اصلی می‌باشد که به جهت رفع این تناقض در راهکار مطرح می‌شود راهکار اول **معمولی (multiprogramming)** می‌باشد یعنی معنای واقعی CPU را واسطه‌ای برای فرآیند (Process) برنامه‌های درون حافظه قرار می‌دهد.

نیاز داشته‌اند به فرآیند دیگری که نتوانستند فرآیند فعلی را به

روز جهانی کار و کارگر

تا بر زلف تو در دست نسیم افتادست

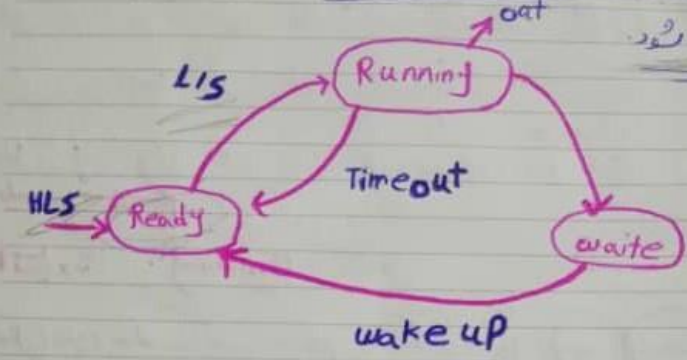
دل سودازده از غصه دو نیم افتادست

۱	۲	۳	۴	۵	۶
۷	۸	۹	۱۰	۱۱	۱۲
۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
۲۵	۲۶	۲۷	۲۸	۲۹	۳۰

دوره wait منتقل می کند و فرکانس جدیدی را اجرا می نماید

تا زمانی که برای فراگرفته شدن wake up تلفظ داده می شود

در درون حافظه قابل شود



۱۲ حافظه کش برای نوع این مشکل از حافظه کش یا Cache memory بدون CPU اتفاق می افتد

۱۳ به این صورت که اگر CPU به داده ای نیاز داشته باشد ابتدا داده مورد نظر را از درون حافظه کش

موردات قاره فراهم می کند در صورت معصوم بودن داده یک بیت امضا یا پر فورد داده مورد نظر

۱۴ داده حقوق به همراه داده های صورتش از درون حافظه کش miss رخ دهد

داده حقوق به همراه داده های صورتش از درون حافظه کش miss رخ دهد

مشکل و تعداد زیر امکان است چرا که برای هر یک داده های جدا

داده های مغلی CPU نیاز به الفانیت به پر فورد داده های مغلی

CPU درون حافظه کش می تواند از طریق رابطه زیر

$$\text{hit ratio} = \frac{\text{hit}}{\text{hit} + \text{miss}}$$

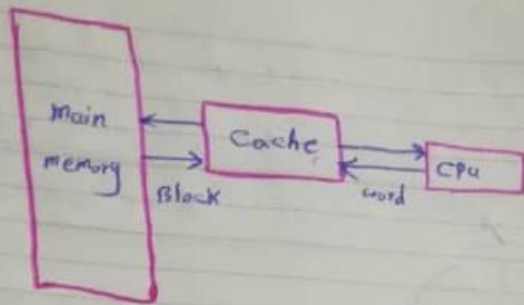
باز به شود

نکته: نرخ انتقال اطلاعات با CPU و حافظه کش word با هم



باز به و با این حافظه کش و حافظه اصلی مورد فواید خود

M	T	W	T	F	S
	1	2	3	4	5
6	7	8	9	10	11
13	14	15	16	17	18
20	21	22	23	24	25
27	28	29	30	31	



**طبیعه سیستم
انتیاده Binding**

فصل های مختلف یک سیستم می باشد که در آن زبان (دوره) پیاده سازی می شود

انتیاده های که در زبان تولید می شوند بعد از ساختارهای زبان برای پیاده سازی و مشخص کردن نحوه ساختارهای مختلف داده ها و انواع ساختارها و اتصال های مختلف دستورات

در این زبان مقیاس داده بندی

در زبان پیاده سازی زبان در زمان (مورد به بخش اعداد و عملیات) کامپایلی مشخص خواهد شد

در زمان ترجمه زبان انتیاده را می توان بر اساس مقیاس بندی خود (۱۰٪) انتیاده های مختلف

بر اساس نوع تعیین می شود - انتیاده های که توسط مقیاس مشخص خواهد شد انتیاده های که توسط

بارگذاری (Loaders) مشخص خواهند شد

انتیاده های که توسط مقیاس تعیین می شود با انتیاده های مشخصه انتیاده های که توسط مقیاس

می شود مانند آدرس های عباری (format) برای مقیاسها و انتیاده های که توسط بارگذاری مشخص خواهد شد

می توان آدرس آدرس های عباری در آدرس های واقعی در زمان حافظه اعلان می باشد

انتیاده های زبان ۴۴ اجرا انتیاده های خواهد بود که در ضمن ورودیها به زبان یا بلوک خاصی

در برنامه مطرح می شود و نوع دیگر انتیاده زبان اجرا در هر نقطه ای از



۶	۵	۴	۳	۲	۱	
۱۳	۱۲	۱۱	۱۰	۹	۸	۷
۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴
۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱
			۳۱	۳۰	۲۹	۲۸

برنامه ها تا نقطه ای که مقدار آنها به مقدار یک برسد متوقف

مثال در دست $x=10$

به عنوان مثال برنامه ای ساده بنویسید که در هر بار از ورودی دریافت خود عددی را بگیرد و تا آنجا که آن عدد را یکبار در هر بار از ورودی دریافت خود عددی را بگیرد.

```
#include <iostream>
```

```
float sum (float, float); // Declaration
```

```
int main () {
```

```
float x, y;
```

```
cin >> x >> y; Actual parameter
```

```
cout << sum (x, y); // call
```

```
return (0); }
```

```
float sum (float a, float b) { // Define
```

```
return (a+b);
```

format parameter

حال چگونه در برنامه فوق مشخص شده است پارامترهای که در فراخوانی تابع ظاهر می شوند پارامترهای

واقعی و پارامترهای که در تعریف تابع نمایان می شود پارامترهای معاری خواهند بود (a, b)

و اولین اتفاق هر دو فراخوانی یک بر سر برنامه مطرح می شود تا مطمئن باشیم این خود دست پارامتر

خواهد بود که تا مطمئن فوق در زبان های برنامه معاری به در صورت تا مطمئن و تا مطمئن با نامی باشند

که در حالت تا مطمئن اولین پارامتر واقعی با اولین پارامتر معاری



روز بزرگداشت شیخ صدوق اردیبهشت ماه دومین پارامتر معاری و سومین پارامتر واقعی با دومین پارامتر معاری و سومین پارامتر واقعی

شعباد خانه پرور ما از که کمتر است باغ چرا چه حاجت سرو و صنوبر است

بارزتر و قوی با n این بارزتر سازی متناظر با تعداد در مثال فوق

بارزتر x با بارزتر y و بارزتر y با بارزتر x متناظر خواهد شد که این

زبان های برنامه سازی از متناظر موقعیتی اتفاق دهی نماید و متناظر با نام بارزتر های واقعی و معنایی بر اساس نام مثال متناظر خواهد شد.

مثال: انقادهای مطرح به همراه زبان و مقدمات رایج برای دستورات $K8 = K + 2$

۱- نوع تغییرات در زبان ترجمه زبان

۲- مقدار تغییرات در زبان اجرا

۳- مثال تلازمی عملگر حسابی جمع در زبان تعریف زبان

۴- عملگر جمع در زبان ترجمه زبان که این عملگر در زبان با عملگر جمع در زبان تعریف می شود و به حسب این که عملگرهاش از چه نوع باشد امکان مختلفی

را ایجاد می کند. در زبان فوق آنکه عملگرهای عملگر جمع عددی صحیح باشند عمل جمع صحیح و

آنکه از نوع اعشاری باشند عمل جمع اعشاری که از نوع صحیحی باشند عمل اجتماع دو مجموعه

و آنکه از نوع رشته ای باشند عمل الفبا جودسته و الفبا صفا در آن بنا بر این زبان

فوق باستی در زبان ترجمه زبان نوع عملگرهای این عملگر مشخص شود تا تا بنا بر این

زبان بتواند عملگر مناسب را داشته باشد

۵- مثال تلازمی عدد ۲۰ در تعریف زبان و جزئیات مربوط به آن مانند تعداد بیت های

اقتصادی در زبان پیاده سازی زبان مشخص خواهد شد

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰
۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰



المنت لله که در میگذره باز است

زان رو که میا بر در او روی نیاز است

۶- محدوده مقادیر معادله برای تغییر k در زمان t یا در زمان t یا در زمان t

$f \text{ bat } a ;$

تست: دستور بدست در زمان t در نظر بگیرید

$a = a + \Delta a$

انفکاد دفع مقید و مجموع مقادیر ممکن مقید a برتر است در هر طریقی ممکن می‌شود

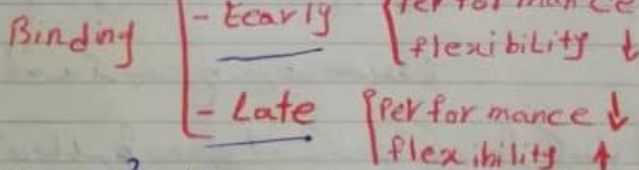
الف) ترجمه زبان - تلفظ زبان

ب) ترجمه زبان - یادده سازی زبان

ج) ترجمه زبان - ترجمه زبان

د) یادده سازی زبان - تلفظ زبان

انواع انقیاد: انقیاد می‌تواند به دو صورت زودرس و دیررس مطرح شود



انقیاد زودرس در زمان ترجمه مطرح می‌شود ولی انقیاد دیررس در زمان اجرا مشخص خواهد شد

در انقیاد زودرس با توجه به اینکه در زمان کامپایل ترجمه صورت می‌گیرد دارای کارایی بالا می‌باشد

انحطاف پذیری پایین خواهد بود در انقیاد دیررس با توجه به اینکه در زمان اجرا انجام می‌شود

دارای کارایی پایین ولی انحطاف پذیری بالا خواهد بود



M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

به باغ جنگ مخور می که محتسب نیز است

اگر چه باده فرج بخش و باد گل‌بیز است

۹۰۰۰

8 May Wed 2019

۱۸

۳ رمضان ۱۴۴۰

چهارشنبه اردیبهشت

۹۸/۲/۸۸ هفته ۵۴۹ | ۰۳۱۶

الف) در راهی انقراض زبان ها کدام فرقیه صحیح است؟
 ۰۸
 خ) در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری در سرعت اجرای برنامه مناسب تر است
 ۰۹

ب) در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری بیشتر ولی مقید سازی زود هنگام سرعت اجرای برنامه ها مناسب تر است.
 ۱۰

ج) در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری کم تر ولی مقید سازی زود هنگام سرعت اجرای برنامه ها مناسب تر است.
 ۱۲

د) در زبان های با مقید سازی زود هنگام قابلیت انعطاف پذیری در سرعت اجرای برنامه ها مناسب تر است.
 ۱۳

طرحه هفتم

۱۴ انواع مفعولها: برای اینکه برنامه نوشته شده در یک زبان بر نامه سازی به زبان قابل معم برای سیستم کامپیوتری ترجمه شود مفعولهای مختلفی است اینها این محل درگیر می باشند که به شرح زیر خواهد بود:

۱۶ ۱- کامپایلر: معصومی است که معمولاً یک زبان سطح بالا را به زبانی نزدیک به زبان ماشین واسطی تبدیل می نماید تا بر این می توانیم به معنی کامپایلر به معنای می توانند یک برنامه سطح بالا را به زبان ماشین واسطی (اوه) تبدیل نمایند.

۱۸ ۲- Assembler: معصومی است که در زبان صدا آن زبان ماشین واسطی و زبان مقید آن زبان ماشین واسطی (اوه) می باشد.

→	۱	۲	۳	۴	۵	۶
→	۷	۸	۹	۱۰	۱۱	۱۲
→	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
→	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
→	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰
→	۳۱	۳۲	۳۳	۳۴	۳۵	۳۶

روز بیماری های خاص و ضعف العلاج

حال دل با تو گفتم هوس است

خبر دل شنفتم هوس است



۹۰۵۸

8 May Wed 2019

۱۸

۳ رمضان ۱۴۴۰

چهارشنبه اردیبهشت

۹۸۰۲۸۸ هفته ۵۴۹۱۸-۲۱۶

۸۸ **الف** در راهی انقضاء زبان ها هنگام فریزه جمع است ؟

۸۹ **الف** در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری در سرعت اجرای برنامه مناسب تر است

۹۰ **ب** در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری بیشتر ولی مقید سازی زود هنگام سرعت اجرای برنامه ها مناسب تر است.

۹۱ **ج** در زبان های با مقید سازی دیر هنگام قابلیت انعطاف پذیری کم تر ولی مقید سازی زود هنگام سرعت اجرای برنامه ها مناسب تر است.

۹۲ **د** در زبان های با مقید سازی زود هنگام قابلیت انعطاف پذیری در سرعت اجرای برنامه ها مناسب تر است.

طرحه هشتم

۹۳ **انواع معضرها** برای اینکه برنامه نوشته شده در یک زبان بر پایه سازی به زبان قابل معمم برای سیستم کامپیوتری ترجمه شود معضراتی متعلق است انجام این عمل در کامپیوتر می باشد که به شرح زیر خواهم نمود :

۹۴ **۱- کامپایلر** معضری است که معمولاً یک زبان سطح بالا را به زبانی نزدیک به زبان ماشین واسطی تبدیل می نماید تا بر این می توانیم معضره کامپایلر به نامی می توانند یک برنامه سطح بالا را به زبان ماشین واسطی (اده) تبدیل می ی.

۹۵ **۲- Assembler** معضری است که زبان صبا آن زبان ماشین واسطی در زبان معقد آن زبان ماشین واسطی (اده) می باشد.

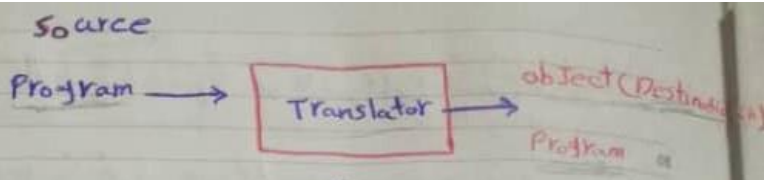
۱	۲	۳	۴	۵	۶
۷	۸	۹	۱۰	۱۱	۱۲
۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
۲۵	۲۶	۲۷	۲۸	۲۹	۳۰
۳۱	۳۲	۳۳	۳۴	۳۵	۳۶



روز بیماری های خاص و ضعف اعصاب

حال دل با تو گفتم هوس است

خیر دل شنفتم هوس است

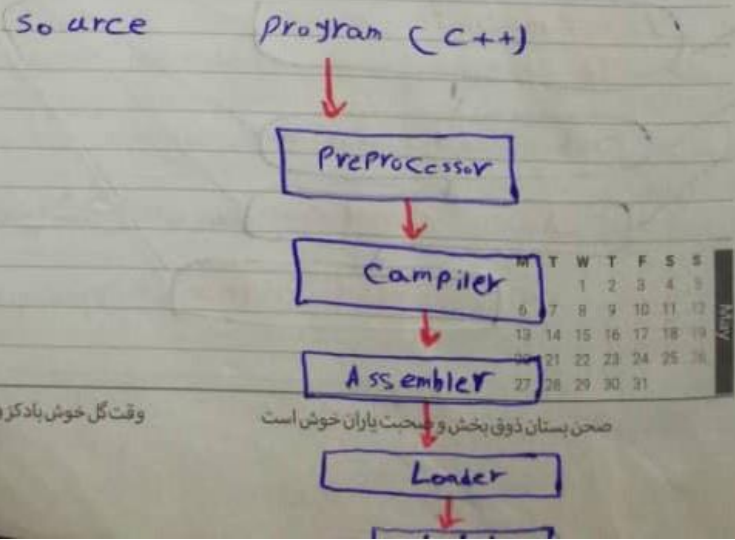


۳- پیش پردازنده یا پیش پردازشگر (preprocessor) نوعی مفسر است که زبان مبدأ و مقصد آن یکسان است به عنوان مثال در تعریف include دستبندی برای های ++C و آنرا سرزنه می گویند و مقصد این دستورات زبان ++C می باشد.

۴- بارنده یا Loader: آدرس تولید شده توسط مفسر را میگیرد و توسط CPU برای مفسرهای مختلف برنامه آدرس های مجازی می باشد معنای این آدرس ها ممکن است توسط واحد حافظه قابل رویت باشد از این رو مفسر بارنده این آدرس را در آدرس های مجازی تولید شده را به آدرس های واقعی درون واحد حافظه منتقل نماید.

۵- Linker: وظیفه این مفسر ایجاد ارتباط میبندد مابین مفسرهای مختلف یک برنامه می باشد تا آن برنامه را در قالب یک برنامه اجرایی واحد تبدیل نماید به عنوان مثال ترتیب استفاده از

مفسر های مطرح شده عمدتاً اجرا می شوند یک برنامه در زبان ++C به صورت زیر می باشد



وقت گل خوش باد کز وی وقت میخواران خوش است

صحن بستان ذوق بخش و صحبت یاران خوش است

کارهای ما می‌تواند هم‌زمان با هم داشته باشد

۱. ترجمه‌کننده از تطبیق معنوی (Lexical Analyser) و تطبیق نحوی (Syntax Analyser)

۲. تطبیق معنایی (Semantic Analyser) و تولید کد میانی یا مابین

۳. بهینه‌سازی Code و تولید کدهای یا Code Generation و منحصراً ما می‌توانیم علاوه بر تشخیص کار

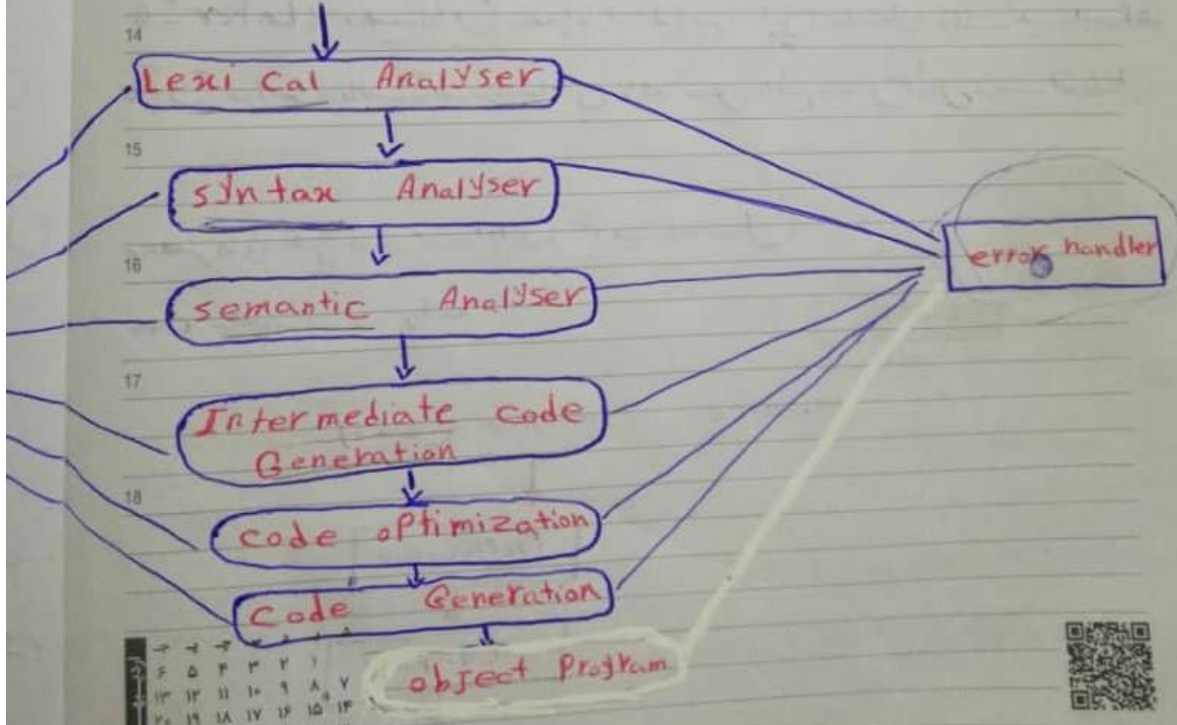
نامیده شده برای درمؤلفه دیگر نیز با توجه به علامت‌ها جدول نماد (Symbol table)

۴. و در آن خطاها را با (error handler) زیرا در صورت از تشخیص کار ما می‌تواند احتمال بروز

خطاها دارد در آن خطاها را تا حد امکان برطرف می‌کند که ما می‌توانیم

برنامه متوقف نشود و برنامه‌نویس زبان می‌تواند خطاها را در نظر دد

COURCE Program



۱	۲	۳	۴	۵	۶
۷	۸	۹	۱۰	۱۱	۱۲
۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
۲۵	۲۶	۲۷	۲۸	۲۹	۳۰

به صد هزار زبان بلیش در اوصاف است

کنون که بر کف گل چار بناده صاف است

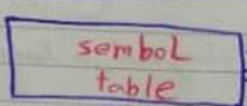


تخلیله نحوی

این دستاوردی از کامپایلر محسوب می شود. اردیبهشت یکشنبه
در بسیاری نوشته شده در طول زبان برنامه سازی دستبندی دارد و بسیار نوشته شده را بصورت

۱۰ **تخلیله نحوی** به این معنی است که خود را از بین خطوط بیابا پس شده Token یا کارهای اساسی می گایه
یعنی از کارهای اساسی شده برای زبان معیار معین مشخصی دارد مانند لغات لغوی زبان یا
Keyword و مابقی کارهای اساسی شده توسط برنامه نویس خواهد بود مانند اسمی
متغیر ثابت ها و به کارهای اساسی شده یک مناسب به صورت یک اختصاص می دهد و در جدول
کار در صفر کرده تا ما: های عددی و به کارهای عددی عملی تمام از آن ها استفاده می گایه این
تخلیله در بین بیابا پس خطوط برنامه از کامپایلر یا تو فرقیات صرف نظر می کند.

۱۱ **تخلیله نحوی** در این فاز کامپایلر محسوب می شود که دردی این تخلیله فروری تخلیله لغوی
خواهد بود و برای آن ساختار سلسله مراتبی به نام درخت تجزیه **Parse Tree** می سازد و درخت
تجزیه ساختار نحوی برنامه را نشان می دهد و در این فاز کامپایلر ترتیب قرارگیری عبارات در طول برنامه
مورد بررسی قرار می گیرد.



۱۲ **تخلیله معنایی** یا **Semantic** معنی فاز کامپایلر محسوب می شود
که فروری تخلیله نحوی ورودی آن خواهد بود این تخلیله معنایی مختلفی را می گوید دارد
۱ **تخلیله معنایی** نوعی و بررسی می باشد که در دسته بارمترهای واقعی و معنایی
در هنگام فرافزایی یک تابع منظور از کنترل نوع این است که آیا عملوند های یک عملگر با نوع خود
انتقال می یابند یا نه این است یا غیر بر عنوان مثال فرض کنید دست دوم **۹+۹**
خواهد بود و در کنترل نوع هم این دست دوم بدین معنایی که

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		



جزوه درسی طراحی و پیاده سازی زبان های برنامه سازی استاد عسکری - کارشناسی کامپیوتر

و در بسیاری از عملیات بازنمایی و نوع آن ها مناسب
 با انواع دیگر برای نامگذاری آن در این جا باشد یا غیره
 این دستور در زبان ++C بر نوع لا به صورت float و نوع لا به صورت int در نظر

گرفته شده و با توجه به این انواع و از آن گاهی که ملازمه عمل جمع و جمع و جمع اعتباری جدول
 سیستم کامپیوتری معادلاتی که باشد این عمل قابل انجام خواهد بود تا به این نقطه میانی
 می تواند انجام عمل جمع بر نام را متوقف نماید تا آنکه عمل تبدیل نوع انجام عمل تبدیل
 نوع می تواند به دو صورت محدود کننده و ارتقا یافته انجام شود که در مثال فوق شوا داده است را

از نوع float به نوع int تبدیل خواهیم محدود کننده خواهد بود که مطلوب نمی باشد زیرا فرض
 نماید که مقدار ۱۲.۴ را قبل از دستور به صورت $x = ۱۲.۴$ داشته باشیم خواهیم تبدیل
 نوع محدود کننده را به عددی آن امکان انجام عمل به مقدار ۱۲.۴ به مقدار ۱۲ محدود می شود و ۰.۴

از داده اولیه لا از این خواهد رفت که ممکن است این ۰.۴ را برای برنامه نویسی
 زبان ++C فرض باشد و اگر تبدیل نوع نخواهد به عکس این حالت انجام دهد تبدیل نوع کمتر
 یا ارتقا یافته خواهد بود به همان تبدیل و از int به float و خود عمل تبدیل نوع می تواند به دو صورت

صریح و ضمنی انجام شود که تبدیل نوع صریح به کمک عملگر $static_cast$ سری توابع از پیش ساخته شده زبان
 توسط برنامه نویسی صورت می گیرد و در تبدیل نوع صریح توسط فار تبدیل میانی میانی
 از دید برنامه نویسی می توان صورت می گیرد و معمولاً به صورت ارتقا یافته خواهد بود

تولید کننده **مینی یا جی** از این کار نامیده می شود به بعد برنامه نویسی می تواند تبدیل خواهد شد
 و در ارتقا یافته معمولاً مورد بررسی قرار می گیرد در این کار نامیده می شود که میانی تولید می شود
 تا به نام نوشته شده با توجه به این که میانی در سطح میانی ما این برنامه میانی فاصله

۶	۵	۴	۳	۲	۱
۱۳	۱۲	۱۱	۱۰	۹	۸
۲۰	۱۹	۱۸	۱۷	۱۶	۱۵
۲۷	۲۶	۲۵	۲۴	۲۳	۲۲
			۳۰	۲۹	۲۸



جزوه درسی طراحی و پیاده سازی زبان های برنامه سازی استاد عسکری - کارشناسی کامپیوتر

بهینه سازی کدها همانطور که می دانیم با توجه به اصول طراحی

الگوریتم بهینه سازی برای کدها می تواند به دو صورت زمان و حافظه

و حافظه ای صورت بگیرد در حالت بهینه سازی زمانی یک کد معادل برای کد اولیه مطرح

می شود پس که بدون تغییر عملکرد آن دارای تابع زمانی یا $T(n)$ مطلوب تری باشد

و در حالت بهینه سازی حافظه ای برای این کدها تعاضیم می دهیم که معادل یا بسیم که بدون

تغییر عملکرد آن معروف حافظه کمتری داشته باشند

تولید کد نهایی: این فاز از خروجی فاز بهینه سازی کد است که نهایی تولید می نماید که بعد از برای

کامپایلر ها که نهایی تولید شده زبان ماشین را تبدیل خواهد نمود

مثال: با فرض اینکه تغییرهای C, B, A هر سه از نوع صحیح سازد باشند خروجی شش ما را می بینیم

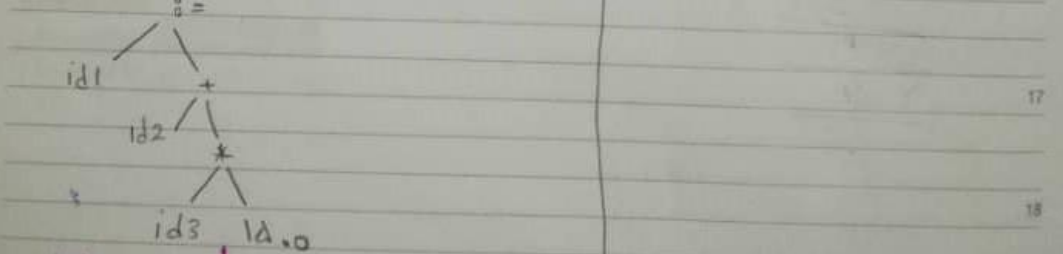
$A_i = B + C * 10_i$

برای دستور $A_i = B + C * 10_i$ تا بسیم بهسیم

code optimization
 $id1 = id2 + id3 * 10_i$

Temp1 = id3 * int_to_real(10)
 $id1 = id2 + temp1$

code generation



intermediate code generation

```
Temp1 = int_to_real(10)
Temp2 = id3 * temp1
```

```
Temp3 = id2 + Temp2
id1 = Temp3
```

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		



لغو اعتبار کتابکوبه شورای آیت الله میرزا حسن شیرازی ۱۳۷۷ هـ.ق

دری دیگر زدن اندیشه تبه دانست

به کوی میگذر هر سالی که ره دانست