

«به نام خدا»

در این جلسه نیز در ادامه‌ی مطالب جلسه پیش ، سعی می‌کنیم کمی بیشتر با نمونه برنامه نویسی در زبان C آشنا شویم .

متغیر چیست ؟

متغیر قسمتی از حافظه است که ما برای آن یک نام دلخواه انتخاب می‌کنیم و از آن برای نگه داری اطلاعات مورد نیاز خود در روند اجرای برنامه استفاده می‌کنیم .

۱- متغیرها با فاموش شدن مدار پاک می‌شوند و حافظه دائمی نیستند .

۲- باید نوع اطلاعاتی که قرار است در متغیر نگه داری شود ، معلوم گردد ، مثلاً قرار است در آن عدد ذخیره شود یا مروف ، یا عدد اعشاری یا

۳- کامپایلر به صورت خودکار بخشی از حافظه را به متغیر مورد نیاز ما اختصاص می‌دهد و نیازی نیست ما برای آن مشخص کنیم که اطلاعات را در کجای حافظه ذخیره کند . البته می‌توان در صورت نیاز آدرس بخشی از حافظه را مشخص کرد تا اطلاعات ما در آن جا ذخیره شود (که فعلاً به آن نمی‌پردازیم) .

تعریف متغیر

برای تعریف یک متغیر ابتدا باید نوع یا (Type) اطلاعاتی که قرار است در آن ذخیره شود ، نوشته شود ، و بعد با یک فاصله (Space) نام متغیر نوشته شود . به مثال زیر دقت کنید :

```
int a ;
```

در اینجا متغیری با نام "a" و از نوع **integer** یا همان عددی تعریف شده است ، یعنی در این متغیر فقط می‌توان یک عدد صحیح (غیر اعشاری) را ذخیره کرد .

✓ **نکته** : اگر یک عدد اعشاری در آن ریخته شود ، بخش اعشاری آن حذف می‌شود .

هر متغیر از جنس `int` ، دو بایت حافظه را به خود اختصاص می دهد و می توان در آن اعداد در گستره ی `۳۲۷۶۷` تا `۳۲۷۶۸` - را ذخیره کرد .

برای ذخیره سازی مروف (Character) باید متغیر از نوع `Char` تعریف شود . متغیرهای `Char` یک بایت حافظه را به خود اختصاص می دهند و در آن ها می توان تنها یک حرف را ذخیره سازی کرد . مروف را در حافظه به شکل کد شده ذخیره می کنند ، نام سیستم کد کردن مروف کد اسکی (ASCII code) است .

در جدول زیر چند نوع داده (Type Data) ی دیگر نیز معرفی شده است .

Type	اندازه (size)	بازه ی تمت پوشش
Long int	۴ بایت	۲۱۴۷۴۸۳۶۴۷ تا - ۲۱۴۷۴۸۳۶۴۸
Unsigned long int	۴ بایت	۰ تا ۲۱۴۷۴۸۳۶۹۵
Float	۴ بایت	برای اعداد اعشاری
Unsigned int	۲ بایت	۰ تا ۶۵۵۳۵

برای ذخیره سازی اطلاعات در داخل متغیرها نیز از همان عملگر "=" استفاده می کنیم . مثلاً :

```
sum1=۷۵;
```

می توانیم متغیرها را در همان موقع تعریف مقدار دهی کنیم . به این کار مقدار دهی اولیه یا "Initialize" می گویند . مثلاً :

```
int sum1=۷۵;
```

اطلاعاتی که در داخل متغیرها ذخیره می شود ثابت نیست و می توان در هر جای برنامه که لازم بود ، مقدار دیگری در متغیر ذخیره کرد . مثلاً :

```
int Cross1=۳۴;
```

.

- .
- .

Cross1= ۶۸;

اگر بخواهیم مقدار متغیر ثابت و غیر قابل تغییر باشد باید قبل از تعیین نوع متغیر ، کلمه ی "const" را بنویسیم . مثلاً

Const float pi= ۳ .۱۴;

می توان چند متغیر را با هم تعریف کرد و آنها را مقدار دهی کرد . مثلاً :

char a1='a', a۲, a۳, a۴='B';

✓ توجه : برای مقدار دهی متغیرهایی که از جنس "char" تعریف می شوند ، باید مقدار در داخل ' ' قرار بگیرد ، به مثال بالا دقت کنید .

قوانین نام گذاری شناسه ها (Identifiers) در زبان C

شناسه ها همان نام هایی هستند که برای متغیرها ، توابع و ... انتخاب می شوند .

برای انتخاب یک شناسه فقط می توانیم از مروف زیر استفاده کنیم :

۱- اعداد ۹ تا ۰

۲- مروف Z تا a (مروف کوچک)

۳- مروف Z تا A (مروف بزرگ)

۴- زیرخط / underline " _ "

۵- علامت \$

به غیر از این کاراکترها مجاز به استفاده از هیچ کاراکتر دیگری (متی فاصله (Space)) نیستیم .

همچنین در ابتدای شناسه ها نمی توانیم از اعداد استفاده کنیم . مثلاً شناسه `loop` غلط است ، ولی `loop۲` درست است .

طول شناسه ها نیز نمی تواند بیش از ۳۲ کاراکتر باشد .

بعضی کلمات در این زبان جزو کلمات رزرو شده (Reserved word) هستند و نمی توانند به عنوان شناسه استفاده شوند مانند : `int, float, void, char, while, if` ، و ...

نکات مهم در مورد برنامه نویسی در زبان C

۱- در پایان هر دستور باید یک ";" گذاشته شود .

۲- جملات و عبارات غیر عددی را باید داخل " " قرار دهیم . مثلاً اگر می خواهیم کارکتر B را در داخل متغیری با نام Temp که از جنس char تعریف شده است ذخیره کنیم ، باید بنویسیم :

```
Temp='B';
```

۳- زبان C در اصطلاح یک زبان Case sensitive است ، یعنی در این زبان بین حروف بزرگ و کوچک تفاوت وجود دارد . مثلاً در یک برنامه ما می توانیم دو متغیر با نام های "temp" و "Temp" داشته باشیم که ارتباطی هم با یکدیگر ندارند .

۴- اگر بخواهیم در هر قسمت از برنامه توضیحاتی را بنویسیم ، باید یک "//" در ابتدای جمله بنویسیم . مثلاً :

```
int a; // etelaate porte C dar in moteghayer rikhte mishavad
```

همچنین اگر بخواهیم چند خط پشت سر هم را موقتاً از روند اجرای برنامه حذف کنیم ، باید علامت "/*" را در ابتدا ، و "*/" را در انتهای آن خطوط قرار دهیم . هرگاه این ۲ علامت را پاک کنیم ، دوباره آن قسمت ، به روند اجرای برنامه اضافه می شود . این روش هم برای افزودن توضیحات به برنامه کاربرد دارد .

۵- در سافتار زیر ، هر دستور یا دستوراتی که در داخل {} نوشته شود ، بی نهایت بار انجام می شود . در حقیقت `while(۱)` ، یک حلقه ی بی پایان است که دستورات داخل آن تا وقتی که مدار فعال باشد ، تکرار می شود . در جلسات آیند شما با سافتار حلقه ها بیشتر آشنا خواهید شد .

```
while(۱)
```

```
{
```

```
PORTD .۳=PIN A .۲;
```

```
PORTD .۴=PIN A .۳;
```

```
}
```

این ۲ دستور مکرراً تا زمانی که میکروکنترلر فعال باشد ، اجرا می شوند .

در بخش برنامه نویسی مطالب بسیار گسترده ای برای آموزش هست ، ولی جلسه ی آینده سعی می کنیم کمی هم از سایر بخش های نرم افزاری میکروکنترلر یعنی همان Code vision صحبت کنیم تا تنوعی هم در مطالب ارائه شده داشته باشیم .

آموزشهای رباتیک طبقه بندی شده توسط کمیته مهندسی رباتیک / nrec.ir (طرح ساماندهی آموزش رباتیک در

اینترنت) برگرفته از سایت رشد مخصوص رده سنی ۱۳ تا ۲۵ سال

گردآوری و ویرایش اولیه : فانم فرناز عطاءاللهی

ویرایش علمی و گرافیکی نهایی : زهره دارابیان



فروشگاه عرضه قطعات الکترونیک ، مکانیک و رباتیک

RoboChip.ir