

در ابتدای برنامه سرآیند `iostream` بارگزاری شده است که برای کار با `cin` و `cout` لازم است. در خط ۲، از فضای نام `std` برای استفاده از `cin` و `cout` و به خاطر کاهش کدنویسی استفاده^۱ شده است. خط ۳ تعریف تابع اصلی^۲ و خط ۱۱ پایان آن است. علت تعریف تابع `main` با نوع داده‌ی `int` به خاطر مشخص کردن نتیجه‌ی برنامه است. اگر برنامه به برگرداند یعنی کار به خوبی انجام شده است. هر عددی به جز ۰ نشانگر وجود خطا و کد ارور است. این مورد در سیستم عامل‌های یونیکس، نتیجه‌ی اجرای برنامه را مشخص می‌کند. برای اطلاعات بیشتر، مطالعه کنید:

<https://www.oreilly.com/library/view/linux-shell-scripting/9781785881985/7d3e28d2-08bf-43d6-a55d-8125446019fe.xhtml>

در خط ۴ دو متغیر به نام‌های `t` و `n` از نوع عدد صحیح تعریف شده‌اند که متغیر `t` در خط ۵ از طریق ورودی، مقدار می‌گیرد.

در خط ۶ یک حلقه تعریف می‌شود که شرط پایان آن، فالسی شدن `t` است؛ در اینجا فالسی شدن `t` وقتی اتفاق می‌افتد که مقدار آن با ۰ برابر شود. بعد از هر بار بررسی شرط، از `t` یک واحد کم می‌شود. این خط معادل شبه کد زیر است:

```
while:
    if(t)
        t = t - 1
        // codes inside of while loop
        goto while
    else
        // rest of code
```

نکته این شرط این است که از هیچ یک از عملگرهای منطقی استفاده نشده است. در اینجا؛

```
if(t) {
    // do something
}
```

همیشه مقدار `t` را بررسی می‌کند. اگر مقدار `t` برابر با ۰ باشد، ارزش متغیر `t`، فالسی^۳ می‌شود و به همین خاطر شرط برقرار نمی‌شود ولی اگر مقدار `t` هر چیزی به جز ۰ باشد، ارزش آن غیر فالسی می‌شود و شرط برقرار است. برای مطالعه در مورد مقدارهای دیگری که ارزش یک متغیر را فالسی می‌کنند، این لینک را مطالعه کنید:

<https://developer.mozilla.org/en-US/docs/Glossary/Falsy>

در خط ۷ وارد حلقه می‌شویم. در اینجا از کاربر یک ورودی گرفته می‌شود و در متغیر `n` قرار داده می‌شود. در خط ۸ عملیات تبدیل نوع^۴ صورت گرفته است و مقدار متغیر `n` به کاراکتر متناظر آن در جدول اسکی تبدیل شده است و نتیجه، چاپ می‌شود. مثلاً اگر کاربر در ورودی عدد ۶۵ را داده باشد، در این خط ابتدا متناظر عدد ۶۵ در جدول اسکی محاسبه می‌شود که برابر با کاراکتر `A` است و سپس `A` چاپ می‌شود. در خط ۱۰ نیز مقدار ۰ بازگشت داده می‌شود تا مشخص شود که برنامه به خوبی پایان یافته است.

1 Using
2 Main
3 Falsy
4 Cast

طبق این موارد، می‌توان نتیجه گرفت که این کد مربوط به یک برنامه‌ی تبدیل اعداد به متن است. برنامه، اول از کاربر تعداد کاراکترهایی که قرار است وارد شوند را می‌پرسد. بعد به ترتیب، مقدار عددی هر کاراکتر را می‌گیرد و به خود کاراکتر تبدیل می‌کند. پس اگر ورودی برنامه این مقادارها باشد:

3

65

66

67

خروجی برنامه برابر با

ABC

خواهد بود.