

تمارین عملی

۱. در این سوال می‌خواهیم ماشین پشته‌ای^۳ را با استفاده از اسمبلی MIPS پیاده‌سازی کنیم. در این سوال مجاز به استفاده از پشته MIPS نیستید. همچنین تعداد pushها مشخص نیست و نمی‌توانید از حافظه ایستا^۴ استفاده کنید. راهنمایی: می‌توانید از sbrk (شماره نهم) برای تخصیص حافظه پویا^۵ استفاده کنید.

این ماشین شامل دستورات زیر می‌باشد:

- psh: زمانی که این دستور می‌آید، در خط بعدی یک عدد صحیح به شما داده می‌شود که این عدد صحیح در پشته push می‌شود.
- pop: این دستور، عدد بالای پشته را pop می‌کند و آن را چاپ می‌کند.
- add: این دستور، دو عدد بالای پشته را pop کرده و حاصل جمع آن‌ها را push می‌کند.
- sub: این دستور، دو عدد بالای پشته را pop کرده و عدد بالاتر را از عدد پایین‌تر کم می‌کند و حاصل تفریق را push می‌کند.
- mul: این دستور، دو عدد بالای پشته را pop کرده و حاصل ضرب آن‌ها را push می‌کند.
- ext: بعد از این دستور، برنامه تمام می‌شود.

نمونه ورودی:

```

1 psh
2 4
3 psh
4 10
5 psh
6 0
7 psh
8 -7
9 sub
10 add
11 mul
12 pop
13 ext

```

نمونه خروجی:

```

1 68

```

توضیح تست: در اینجا ابتدا حاصل $(-7) - 0$ حساب شده که برابر با ۷ است سپس ۷ با ۱۰ جمع شده و در نهایت نیز ۱۷ ضرب در ۴ می‌شود و زمانی که pop می‌کنیم، تنها عدد ۶۸ در استک موجود است و این مقدار نیز pop شده و چاپ می‌شود.

۲. کدی بنویسید که دو عدد a و b را ورودی بگیرد و a^b را به صورت بازگشتی محاسبه کند. در این محاسبه شما باید از پیچیدگی $O(\log(b))$ استفاده کنید و محاسبه توان به صورت خطی مجاز نیست.

۳. برنامه‌ای به زبان اسمبلی MIPS بنویسید که در آرایه‌ای به طول n، تعداد جفت اعدادی که نسبت به هم اول هستند را محاسبه کند.

^۳ Stack Machine
^۴ static
^۵ dynamic

در ورودی، ابتدا عدد n که طول آرایه است داده می‌شود. سپس n عدد که اعضای آرایه هستند ورودی داده می‌شوند. خروجی برنامه تعداد جفت اعدادی که نسبت به هم اول هستند را نشان می‌دهد.

راهنمایی: برای تشخیص اینکه دو عدد نسبت به هم اول هستند یا خیر، می‌توانید از الگوریتم اقلیدسی استفاده کنید.

۴. برنامه‌ای به زبان اسمبلی MIPS بنویسید که ابتدا عدد n را از ورودی دریافت کند. سپس n عدد صحیح را از ورودی روی خطوط مجزا بخواند، و با استفاده از Bubble Sort آنها را از بزرگ به کوچک مرتب کند. سپس آنها را به این ترتیب و در خطوط مجزا در خروجی چاپ کند.
تضمین می‌شود که $n \leq 200$.

۵. برنامه‌ای به زبان اسمبلی MIPS بنویسید که یک ماتریس $N \times M$ را در یک ماتریس $M \times P$ ضرب کند و حاصل را در حافظه ذخیره کند. (می‌توانید فرض کنید اعداد P, M, N و ماتریس‌ها در حافظه ذخیره شده‌اند).
به عنوان مثال:

```
1 .data
2 N: .word 3
3 M: .word 2
4 P: .word 4
5 A: .word 1, 2, 3, 4, 5, 6      # { {1, 2}, {3, 4}, {5, 6} }
6 B: .word 8, 7, 6, 5, 4, 3, 2, 1 # { {8, 7, 6, 5}, {4, 3, 2, 1} }
```

۶. یک عدد آرمسترانگ^۶ عددی است که برابر با مجموع ارقام خود باشد، به طوری که هر رقم به توان تعداد ارقام آن عدد رسیده باشد. برنامه‌ای به زبان اسمبلی MIPS پیاده‌سازی کنید که یک عدد هشت بیتی به عنوان ورودی دریافت کرده و در صورتی که عدد آرمسترانگ بود، YES و در غیر این صورت NO چاپ کند.

۷. برنامه‌ای بنویسید که حاصل ضرب دو چندجمله‌ای را حساب و چاپ کند. در ورودی برای هر چند جمله‌ای ابتدا درجه‌ی آن داده می‌شود و سپس ضرایب آن به ترتیب از ضریب پرارزش تا کم‌ارزش داده می‌شوند. خروجی شما باید یک رشته با فرمت مناسب باشد:

- جملات با ضریب ۰ نشان داده نشوند.
- ضریب ۱ نشان داده نشود.
- اگر جمله‌ی سمت چپ مثبت بود، علامت آن نشان داده نشود.

مثال:

```
1 Input:
2 3
3 1
4 1
5 0
6 5
7 2
8 -2
9 0
10 2
11
12 This input represents (x^3+x^2+5)(-2x^2+2). Output:
13
14 -2x^5-2x^4+2x^3-8x^2+10
```

حداکثر درجه‌ی هر ورودی، ۱۰۰ است. نمونه‌ی ورودی‌ای که کنار برنامه‌ی خود قرار می‌دهید باید رعایت تمام موارد بالا در کدتان را نشان دهد.