Laboratório 3

Revisando laços e tornando os códigos um pouco mais complexos!



Dicas

- Você não precisa entregar nenhum código como resposta. Procure entender os conceitos e explorar as variações.
- Você pode utilizar o simulador RISC-V para testar os códigos que você desenvolver. Para isso, veja as dicas fornecidas logo abaixo.
- Não deixe de colocar comentários nos seus códigos. Procure organizar o código de forma que ele fique mais fácil de entender.

Você sabe detectar um número ímpar?

Quando consideramos números em binário, é bem simples detectar um número ímpar, basta observar o bit menos significativo (o bit mais à direita). Se ele for 1, o número é ímpar, se for 0, o número é par. Veja alguns exemplos: 0101 (número 5) é ímpar, 0110 (número 6) é par, 1001 (número 9) é ímpar, 1010 (número 10) é par.

Para testar um bit (ou vários), você pode utilizar a instrução AND. Ela realiza uma operação E lógica bit a bit, para todos os bits do número. Então, um AND entre os números 6 (0110) e 5 (0101) tem como resposta o número 4 (0100) pois somente os bits com valor 1 em ambos os números foram mantidos como 1 no resultado final.

Logo, para testar se um número é ímpar, basta fazer um AND com o valor 1 (0001). Dessa forma, você está mantendo apenas o último bit, que poderá ser 0 (par) ou 1 (ímpar).



Atividade 1

Faça um programa que leia um número do teclado e imprima a letra I se o número for ímpar ou a letra P se o número for par. Busque os códigos da letra I e P na tabela ASCII para poder imprimir o carecter corretamente.



Atividade 2

Você consegue fazer outra versão do programa que detecte se um número é múltiplo de 4? Nesse caso, imprima S para sim e N para não.



Dica

A instrução OR faz uma operação OU lógica bit a bit, para todos os bits do número. Então, um OR entre os números 6 (0110) e 5 (0101) tem como resposta o número 7 (0111) pois todos os bits com valor 1 em, ao menos, um dos números foram mantidos como 1 no resultado final.

Integrando com laços

É muito comum que testes sejam colocados para decidir trechos diferentes de código a serem executados. Nas atividades anteriores, você conseguiu detectar se um número é ímpar ou se é par e também se ele é múltiplo de 4. Agora, vamos integrar esses testes com laços para que o programa possa ler vários números do teclado e executar tarefas diferentes para cada tipo de número.



Atividade 3

Faça um programa que leia múltiplos números do teclado. Seu programa deve parar quando for digitado o valor 0. Ao final do programa, ele deve imprimir o resultado da soma de todos os ímpares subtraindo da soma de todos os pares. Resultado = Soma(ímpares) - Soma(pares).

Você sabe quardar segredos?

Criptografia é uma forma de codificar informações para que ela só seja legível por pessoas capazes de decodificá-la.

Uma forma bem simples de codificar uma informação é invertendo alguns dos bits de um número e guardar em segredo quais bits foram invertidos. Para fazer isso, a instrução XOR é muito útil. Ela realiza uma operação OU exclusivo bit a bit, para todos os bits do número. Então, um XOR entre os números 6 (0110) e 5 (0101) tem como resposta o número 3 (0011) pois somente os bits com valor 1 em um dos números foram mantidos como 1 no resultado final. Note que os bits com valor 1 em ambos os números foram zerados no resultado final.

Assim, você pode ter um número de segredo que é o XOR entre o número que você quer codificar e um número qualquer. Para decodificar, basta fazer o XOR entre o número codificado e o número de segredo.



Atividade 4

Faça um programa que leia dois números do teclado: o segredo e o número a codificar. O programa deve imprimir o número codificado. Para isso, utilize a instrução XOR. Note que esse programa servirá tanto para codificar quanto para decodificar o número.



Atenção

Apesar de ser uma forma de codificar um número, essa técnica não é segura. Ela é apenas uma forma de você aprender a utilizar a instrução XOR.

E quando as mensagens são maiores?

Você pode aumentar o tamanho das mensagens repetindo a operação múltiplas vezes. Essa é uma das formas de codificar uma mensagem grande.



Atividade 5

Faça um programa que leia o segedo do teclado e, depois, leia vários números do teclado imprimindo cada um codificado em sequência. O programa deve parar quando for digitado o valor 0.

Desafio final

Você aprendeu sobre números binários e agora precisa encontrar uma forma de fazer conversões simples entre números decimais e binários. O método clássico de divisões sucessivas é a forma comum de fazer em papel. Entretanto, você pode fazer isso de forma mais rápida utilizando a instrução de deslocamento de bits (srl, que desloca bits para a direita ou sll que desloca bits para a esquerda).

Você também já sabe como testar um bit específico de um número utilizando a instrução AND. Então, você pode fazer a conversão de um número de decimal para binário utilizando a instrução AND a instrução de deslocamento.



Atividade 6

Faça um programa que leia um número do teclado e imprima o número em binário. Para isso, utilize as instruções AND e de deslocamento de bits. Como os números do seu processador são de 32 bits, você deve ter um laço for no seu código com 32 interações.

Conclusões

Você desenvolveu programas um pouco mais complexos, que manipulam bits para realizar suas tarefas. Você também integrou esses programas em laços para realizar as tarefas mais vezes.



Resumo

Você já sabia que números pares e ímpares são diferentes mas não tinha testado apenas 1 bit para verificálos. E agora já sabe também uma forma simples de codificar um segredo, através da instrução XOR. Além disso tudo, você conseguiu converter números decimais em binário!