NUS National University of Singapore | Computing

NUS WebMail IVLE LIBRARY MAPS

Search search for... in NUS Websites ▼ GO

CodeCrunch

Home | My Courses | Browse Tutorials | Browse Tasks | Search | My Submissions | Logout

CS1010E 16/17S1 Cut the Rod (Question)

Tags & Categories

Tags:

Categories:

Related Tutorials

Task Content

Cut the Rod

Topic Coverage

- Assignment and expressions
- Nested control statements
- Functions and procedures
- One-dimensional arrays

Problem Description

Suppose you have a piece of rod of length 3 units to be cut into several pieces to be sold. Varying lengths of the rod are valued differently. As an example, a rod of length 1 is valued at $3, while that of lengths 2 and 3 are valued at $5 and $8 respectively.

Clearly, a rod of length 1 cannot be cut further; hence the maximum value is $3. A rod of length 2 is valued at $5. Cutting it into two length 1 pieces will fetch $6. Hence the maximum value for a length 2 rod is $6 with one cut. Similarly, a rod of length 3 is valued at $8. Cutting it into one length 1 and one length 2 will fetch $8; however cutting the length 2 piece further will give three length 1 pieces valued at $9. Hence the maximum value for a length 3 rod is $9.

Two other examples are given below for a rod of length 8 with a different pricing table.

- Pricing table #1

Length, n	1	2	3	4	5	6	7	8
Price	1	5	8	9	10	17	17	20

The maximum value of this rod is 22, by having two pieces of lengths 2 and 6.

- Pricing table #2

Length, n	1	2	3	4	5	6	7	8
Price	3	5	8	9	10	17	17	20

The maximum value of this rod is 24, by having eight pieces of length 1 each.

Rather than enumerating all possible ways to cut a rod of a given length and finding the maximum value among the cuts, we adopt the strategy of updating the pricing table beginning with the smallest length. A description of how this is done for the last example is given as follows:

- Rod of length 1 cannot be cut further, hence the price remains at $3.
- Rod of length 2 is originally $5. Cutting it into two length 1 pieces fetches $6. Hence, update the pricing table to be $6.
- Rod of length 3 is originally $8. Using the updated pricing table for lengths 1 and 2, the maximum value is $3+$6=$9.
- Rod of length 4 is originally $9. From the updated pricing table, the maximum value is $12 (either 3+9 or 6+6). Hence, update the pricing table to be $12.
- Rod of length 5 is originally $10. From the updated pricing table, the maximum value is $15 (either 3+12 or 6+9). Hence, update the pricing table to be $15.
- Rod of length 6 is originally $17. From the updated pricing table, the maximum value is $18 (either 3+15 or 6+12 or 9+9). Hence, update the pricing table to be $18.
- Rod of length 7 is originally $17. From the updated pricing table, the maximum value is $21 (either 3+18 or 6+15 or 9+12). Hence, update the pricing table to be $21.
- Rod of length 8 is originally $20. From the updated pricing table, the maximum value is $24 (either 3+21 or 6+18 or 9+15 or 12+12). Hence, update the pricing table to be $24.

Task

Write a program that requests the user to enter the length of the rod `n, 1 ≤ n ≤ 20`, followed by the pricing table for rod lengths `1` to `n`. The program outputs the maximum value of the rod.

Take note of the following:

- Assume that all prices are positive integers.
- Only one sample run is provided to test for format correctness. You should device your own test cases to test your program.

This task is divided into several levels. Read through all the levels (from first to last, then from last to first) to see how the different levels are related. **You may start from any level.**

- **Deadline: Submit your work to CodeCrunch by Thursday, 27 October, 23:59:59.**

Level 1

Name your program `cut1.c`

Write a program that reads in the length of the rod `n` and the pricing table from `1` to `n`. Output the length and pricing table.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
8
1 5 8 9 10 17 17 20
Length: 8
Price of length 1: 1
Price of length 2: 5
Price of length 3: 8
Price of length 4: 9
Price of length 5: 10
Price of length 6: 17
Price of length 7: 17
Price of length 8: 20
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < cut.in | diff - cut1.out
```

To proceed to the next level (say level 2), copy your program by typing the Unix command

```
cp cut1.c cut2.c
```

Level 2

Name your program `cut2.c`

Write a program that reads in the length of the rod `n` and the pricing table from `1` to `n`. Output the length and pricing table. For every length `i` in the pricing table, output the possible ways to cut rod `i`. For example, if `i = 4`, then output `[1,3]` and `[2,2]`. Note that `[3,1]` should not be output.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
8
1 5 8 9 10 17 17 20
Length: 8
Price of length 1: 1
Price of length 2: 5
[1,1]
Price of length 3: 8
[1,2]
Price of length 4: 9
[1,3]
[2,2]
Price of length 5: 10
[1,4]
[2,3]
Price of length 6: 17
[1,5]
[2,4]
[3,3]
Price of length 7: 17
[1,6]
[2,5]
[3,4]
Price of length 8: 20
[1,7]
[2,6]
[3,5]
[4,4]
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < cut.in | diff - cut2.out
```

To proceed to the next level (say level 3), copy your program by typing the Unix command

```
cp cut2.c cut3.c
```

Level 3

Name your program `cut3.c`

Write a program that reads in the length of the rod `n` and the pricing table from `1` to `n`. Output the length and pricing table. For every length `i` in the pricing table, output the possible ways to cut rod `i` and the price associated with such a cut. For example, using the pricing table for a rod of length `8` as shown above, if `i = 4`, then output `[1,3] = 9` and `[2,2] = 10`. Note that `[3,1]` should not be output.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
8
1 5 8 9 10 17 17 20
Length: 8
Price of length 1: 1
Price of length 2: 5
[1,1] = 2
Price of length 3: 8
[1,2] = 6
Price of length 4: 9
[1,3] = 9
[2,2] = 10
Price of length 5: 10
[1,4] = 10
[2,3] = 13
Price of length 6: 17
[1,5] = 11
[2,4] = 14
[3,3] = 16
Price of length 7: 17
[1,6] = 18
[2,5] = 15
[3,4] = 17
Price of length 8: 20
[1,7] = 18
[2,6] = 22
[3,5] = 18
[4,4] = 18
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < cut.in | diff - cut3.out
```

To proceed to the next level (say level 4), copy your program by typing the Unix command

```
cp cut3.c cut4.c
```

Level 4

Name your program `cut4.c`

Write a program that reads in the length of the rod `n` and the pricing table from `1` to `n`. Output the length and pricing table. For every length `i` in the pricing table, output the possible ways to cut rod `i` and the price associated with such a cut. For example, using the pricing table for a rod of length `8` as shown above, if `i = 4`, then output `[1,3] = 9` and `[2,2] = 10`. Note that `[3,1]` should not be output. Output the different ways to cut the rod of length `i`, find the maximum price and replace the corresponding entry of the pricing table.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
8
1 5 8 9 10 17 17 20
Length: 8
Price of length 1: 1
Max price of length 1: 1
Price of length 2: 5
[1,1] = 2
Max price of length 2: 5
Price of length 3: 8
[1,2] = 6
Max price of length 3: 8
Price of length 4: 9
[1,3] = 9
[2,2] = 10
Max price of length 4: 10
Price of length 5: 10
[1,4] = 11
[2,3] = 13
Max price of length 5: 13
Price of length 6: 17
[1,5] = 14
[2,4] = 15
[3,3] = 16
Max price of length 6: 17
Price of length 7: 17
[1,6] = 18
[2,5] = 18
[3,4] = 18
Max price of length 7: 18
Price of length 8: 20
[1,7] = 19
[2,6] = 22
[3,5] = 21
[4,4] = 20
Max price of length 8: 22
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < cut.in | diff - cut4.out
```

To proceed to the next level (say level 5), copy your program by typing the Unix command

```
cp cut4.c cut5.c
```

Level 5

Name your program `cut5.c`

Write a program that reads in the length of the rod `n` and the pricing table from `1` to `n`. Output the maximum price of the rod after trying all possible ways to cut the rod.

```
$ ./a.out
8
1 5 8 9 10 17 17 20
Max price of rod: 22
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < cut.in | diff - cut5.out
```