

# CPSC 1301, Computer Science I Lab Assignment

## Lab 05b Solutions

### Problem 1

```

1  #-----
2  # gambler.py
3  #-----
4
5  import sys
6  import random
7
8  # Accept integer command-line arguments stake, goal, and trialCount.
9  # Run trialCount experiments that start with stake dollars and
10 # terminate on 0 dollars or goal. Write to standard output the
11 # percentage of wins and the average number of bets per experiment.
12
13 stake = int(sys.argv[1])
14 goal = int(sys.argv[2])
15 trials = int(sys.argv[3])
16
17 # Run trialCount experiments that start with stake and terminate on
18 # 0 or goal.
19 bets = 0
20 wins = 0
21 for t in range(trials):
22     # Run one experiment.
23     cash = stake
24     while cash > 0 and cash < goal:
25         # Simulate one bet.
26         bets += 1
27         if random.randrange(0, 2) == 0:
28             cash += 1
29         else:
30             cash -= 1
31     if cash == goal:
32         wins += 1
33
34 print(str(100 * wins//trials) + '%_wins')
35 print('Avg.#_bets:_' + str(bets//trials))

```

### Problem 2

```

1  #-----
2  # factors.py
3  #-----
4
5  import sys
6
7  # Accept integer n as a command-line argument. Write to standard
8  # output the prime factors of n.
9
10 n = int(sys.argv[1])
11
12 factor = 2

```

```

13 while factor*factor <= n:
14     while n % factor == 0:
15         # Cast out and write factor.
16         n //= factor
17         print(str(factor) + '_')
18     factor += 1
19     # Any factors of n are greater than or equal to factor.
20
21 if n > 1:
22     print(n)
23 print()

```

## Problem 3

```

1 #-----
2 # makechange.py
3 #-----
4
5 import sys
6 import random
7
8 # accept two floating point numbers to 2 decimal places, price and tender
9 # where price <= tender, and
10 # calculate the difference between price and tender in coins
11
12 price = float(sys.argv[1])
13 tender = float(sys.argv[2])
14 change = 0
15 dollars = 0
16 quarters = 0
17 dimes = 0
18 nickles = 0
19 cents = 0
20
21 print("the_price_is_%.2f,_you_tendered_%.2f" % (price, tender))
22
23 if price > tender:
24     print("price_is_greater_than_tender,_exiting...")
25     sys.exit(1)
26
27 price = int(price * 100)
28 tender = int(tender * 100)
29 change = tender - price
30
31 #print("DEBUG> the price is %d, you tendered %d, change is %d" % (price, tender, change))
32 dollars = change // 100
33 if dollars > 0:
34     dollars = dollars
35     change -= (dollars * 100)
36
37 #print("DEBUG> the price is %d, you tendered %d, change is %d, dollars is %d" % (price,
38     tender, change, dollars))
39 quarters = change // 25
40 if quarters > 0:
41     quarters = quarters
42     change -= (quarters * 25)
43
44 #print("DEBUG> the price is %d, you tendered %d, change is %d, quarters is %d" % (price,
45     tender, change, quarters))
46 dimes = change // 10
47 if dimes > 0:
48     dimes = dimes
49     change -= (dimes * 10)

```

```

49 #print("DEBUG> the price is %d, you tendered %d, change is %d, dimes is %d" % (price, tender
    , change, dimes))
50 nickles = change // 5
51 if nickles > 0:
52     nickles = nickles
53     change -= (nickles * 5)
54
55 #print("DEBUG> the price is %d, you tendered %d, change is %d, nickles is %d" % (price,
    tender, change, nickles))
56 cents = change
57 #print("DEBUG> the price is %d, you tendered %d, change is %d, cents is %d" % (price, tender
    , change, cents))
58 print("Your change is %d dollars, %d quarters, %d dimes, %d nickles, and %d cents" % (
    dollars, quarters, dimes, nickles, cents))

```

## Problem 4

```

1 #-----
2 # sample.py
3 #-----
4
5 import sys
6 import random
7
8 # Accept integers m and n as command-line arguments. Write to standard
9 # output a random sample of m integers in the range 0...n-1 (no
10 # duplicates).
11
12 m = int(sys.argv[1]) # choose this many elements
13 n = int(sys.argv[2]) # from 0, 1, ..., n-1
14
15 # Initialize perm.
16 #perm = stdarray.create1D(n, 0)
17 #for i in range(n):
18 #    perm[i] = i
19 perm = list(range(0, n))
20
21 # Create random sample in perm[0], perm[1], ..., perm[m-1]
22 for i in range(m):
23
24     # Choose a random integer r between i and n-1.
25     r = random.randrange(i, n)
26
27     # Swap perm[i] and perm[r].
28     temp = perm[r]
29     perm[r] = perm[i]
30     perm[i] = temp
31
32 # Write the results.
33 for i in range(m):
34     print(str(perm[i]), end = ' ')
35 print()

```

## Problem 5

```

1 #-----
2 # couponcollector.py
3 #-----
4
5 import sys
6 import random
7

```

```

8  # Accept integer n as a command-line argument. Write to standard
9  # output the number of coupons you collect before obtaining one of
10 # each of n types.
11
12 n = int(sys.argv[1])
13
14 count = 0
15 collectedCount = 0
16 #isCollected = stdarray.create1D(n, False)
17 isCollected = [None] * (n + 1)
18
19 while collectedCount < n:
20     # Generate another coupon.
21     value = random.randrange(0, n)
22     count += 1
23     if not isCollected[value]:
24         collectedCount += 1
25         isCollected[value] = True
26
27 print(count)
28 for e in isCollected:
29     print(e, end = '_')
30 print()

```

## Problem 6

Write a prime number finder — seive of Eratosthenes

```

1  #-----
2  # isprime.py
3  #-----
4
5  import sys
6  import math
7
8  x = int(sys.argv[1])
9
10 def prime(x):
11     top = int(math.sqrt(x))
12     #print("top is", top)
13     if x in [0, 1]:
14         return False
15     if x == 2:
16         return True
17     for n in range(2, top+1):
18         #print(x, n, x % n)
19         if x % n == 0:
20             return False
21     return True
22
23 result = prime(x)
24 answer = "true" if result else "false"
25
26 print("Is %d a prime number? %s" % (x, answer))

```