

CGS600A: Computational Tools in Cognitive Science

Assignment #3: Python-2

Max marks: 120

Due on/before: 28 Aug.18, 23.59

20-Aug-2018

All programs are to be written in Python.

1. Consider the problem of creating change for a certain amount of money by using the minimum number of coins/notes. So, assuming we have coins/notes of the following denomination 1, 2, 5, 10, 20, 50, 100, 200, 500, 2000 (as we do in India) write a program that reads in a positive number as amount and then outputs the minimum number of notes/coins and their denominations that will make up the amount.

So, if we had to make up the amount 63, we have $63=50+10+2+1$. To make up 3285 we have $3285=2000+500+500+200+50+20+10+5$.

Briefly justify why your program uses the minimum number of notes/coins. You can do this as comments in your code.

[25]

2. A simple encryption system uses a shifting process to hide a message. The value of the shift can be in the range 1 to 26. For example a shift of 20 means that A = U, B = V, C = W, Y = S, Z = T etc. i.e.

First an extra space is added to the end of the string. To make things a little more difficult, spaces within the original text are replaced with QQ before the text is encrypted. Double Q (QQ) was selected because no English word ends in Q or contains QQ. Additionally the coded message is printed in blocks of six characters separated by spaces. The last block might not contain six characters. Develop an algorithm that takes the coded text (less than 100 characters), the shift value and prints the decoded original text. Your algorithm must reject any non-valid value for shift and display an error message "INVALID SHIFT VALUE". Assume all characters are upper case. Here are some sample inputs and outputs:

INPUT: CODED TEXT: "UHINBY LKKQCH HYLKK", SHIFT: 20
OUTPUT DECODED TEXT: "ANOTHER WINNER"

INPUT: CODED TEXT: "RUIJGG EVGGBK SAGG", SHIFT: 16
OUTPUT DECODED TEXT: "BEST OF LUCK"

INPUT: CODED TEXT: "DKSMMW NAMMUK QMM", SHIFT: 29
OUTPUT: INVALID SHIFT VALUE.

[35]

3. A given string is to be encoded as follows. Assume s (length n) is the string to be encoded and $e=""$ is the empty encoding of s at the start. :

1. If n is 1 or less then $e=es$ (es is the concatenation of e and s).

2. If $n > 1$ pick up the middle character of s (i.e. $n//2$ th character where counts start from 1), say c , and append it to e (i.e. $e=ec$).
3. Divide the string into two sub-strings l (sub-string to the left of the middle character c) and r (sub-string to the right of the middle character). That was $s=lcr$.
4. Repeat the above with l and r until s is fully encoded.

For example let $s='abcdef'$ then the table below shows what happens (ϵ means empty string).

Length	String	Mid char	Left	Right	Encoding
6	abcdef	c	ab	def	c
2	ab	a	ϵ	b	ca
3	def	e	d	f	cae
0	ϵ	ϵ	ϵ	ϵ	cae
1	b	b	ϵ	ϵ	caeb
1	d	d	ϵ	ϵ	caebd
1	f	f	ϵ	ϵ	caebdf

So, the encoding is $e=caebdf$. If we decode $e=caebdf$ then we should get $s=abcdef$.

Write functions `encode` and `decode` such that `encode` takes a string as input and produces an encoded string and `decode` takes an encoded string as argument and decodes it to yield the original string.

[30]

4. Assume there is a staircase with N steps to reach a landing. You can take 1-step, 2-steps or 3-steps at a time. In how many possible ways can you reach the landing. Suppose $N=4$ then we have: $1+1+1+1$, $1+2+1$, $2+1+1$, $1+1+2$, $2+2$, $1+3$, $3+1$ - that is 7.

Write a program which reads in N (assume $1 \leq N \leq 20$) and prints the number of ways you can reach the landing. So for $N=4$ the output is 7.

[30]