

Component C. Personalized Project Reference.

Procedure:

i.

```
def getprompt(mode):  
    global prompt  
    while True:  
        numitems=input("Enter the number of "+mode+" you  
would like in the prompt: ")  
        if numitems.isnumeric():  
            break  
        else:  
            print("Please enter a valid number.",'\n')  
    for i in range(int(numitems)):  
        promptlist.append(random.choice(modelist))  
    prompt=' '.join(promptlist)  
    time.sleep(0.5)
```

ii.

```
getprompt(mode)
```

List:

i.

```
words=['a', 'abandon', 'ability', 'able', 'abortion',  
'about', 'above', 'abroad', 'absence', 'absolute',  
'absolutely', 'absorb', 'abuse', 'academic', 'accept',  
'access', 'accident', 'accompany', 'accomplish',  
'according', 'account', 'accurate', 'accuse', 'achieve',  
'achievement', 'acid', 'acknowledge', 'acquire', 'across',  
'act', 'action', 'active', 'activist', 'activity', 'actor',  
'actress', 'actual', 'actually', 'ad', 'adapt', 'add',  
'addition', 'additional', 'address', 'adequate', 'adjust',  
'adjustment', 'administration', 'administrator', 'admire',  
'admission', 'admit', 'adolescent', 'adopt', 'adult',  
'advance', 'advanced', 'advantage', 'adventure',  
'advertising', 'advice', 'advise', 'adviser', 'advocate',  
'affair', 'affect', 'afford', 'afraid', 'after',  
'afternoon', 'again', 'against', 'age', 'agency', 'agenda',  
'agent', 'aggressive', 'ago', 'agree', 'agreement',  
'agricultural', 'ah', 'ahead', 'aid', 'aide', 'aim', 'air',  
'aircraft', 'airline', 'airport', 'album', 'alcohol',  
'alive', 'all', 'alliance', 'allow', 'ally', 'almost',  
'alone', 'along', 'already', 'also', 'alter',  
'alternative', 'although', 'always', 'amazing', 'among',  
'amount', 'analysis', 'analyst', 'analyze', 'ancient',  
'and', 'anger', 'angle', 'angry', 'animal', 'anniversary',  
'announce', 'annual', 'another', 'answer', 'anticipate',  
'anxiety', 'any', 'anybody', 'anymore', 'anyone',  
'anything', 'anyway', 'anywhere', 'apart', 'apartment',  
'apparent', 'apparently', 'appeal', 'appear', 'appearance',  
'apple', 'application', 'apply', 'appoint', 'appointment',  
'appreciate', 'approach', 'appropriate', 'approval',
```

'approve', 'approximately', 'architect', 'area', 'argue',
'argument', 'arise', 'arm', 'armed', 'army', 'around',
'arrange', 'arrangement', 'arrest', 'arrival', 'arrive',
'art', 'article', 'artist', 'artistic', 'as', 'aside',
'ask', 'asleep', 'aspect', 'assault', 'assert', 'assess',
'assessment', 'asset', 'assign', 'assignment', 'assist',
'assistance', 'assistant', 'associate', 'association',
'assume', 'assumption', 'assure', 'at', 'athlete',
'athletic', 'atmosphere', 'attach', 'attack', 'attempt',
'attend', 'attention', 'attitude', 'attorney', 'attract',
'attractive', 'attribute', 'audience', 'author',
'authority', 'auto', 'available', 'average', 'avoid',
'award', 'aware', 'awareness', 'away', 'awful', 'baby',
'back', 'background', 'bad', 'badly', 'bag', 'bake',
'balance', 'ball', 'ban', 'band', 'bank', 'bar', 'barely',
'barrel', 'barrier', 'base', 'baseball', 'basic',
'basically', 'basis', 'basket', 'basketball', 'bathroom',
'battery', 'battle', 'be', 'beach', 'bean', 'bear', 'beat',
'beautiful', 'beauty', 'because', 'become', 'bed',
'bedroom', 'beer', 'before', 'begin', 'beginning',
'behavior', 'behind', 'being', 'belief', 'believe', 'bell',
'belong', 'below', 'belt', 'bench', 'bend', 'beneath',
'benefit', 'beside', 'besides', 'best', 'bet', 'better',
'between', 'beyond', 'big', 'bike', 'bill', 'billion',
'bind', 'biological', 'bird', 'birth', 'birthday', 'bit',
'bite', 'black', 'blade', 'blame', 'blanket', 'blind',
'block', 'blood', 'blow', 'blue', 'board', 'boat', 'body',
'bomb', 'bombing', 'bond', 'bone', 'book', 'boom', 'boot',
'border', 'born', 'borrow', 'boss', 'both', 'bother',
'bottle', 'bottom', 'boundary', 'bowl', 'box', 'boy',
'boyfriend', 'brain', 'branch', 'brand', 'bread', 'break',
'breakfast', 'breast', 'breath', 'breathe', 'brick',
'bridge', 'brief', 'briefly', 'bright', 'brilliant',

'bring', 'broad', 'broken', 'brother', 'brown', 'brush',
'buck', 'budget', 'build', 'building', 'bullet', 'bunch',
'burden', 'burn', 'bury', 'bus', 'business', 'busy', 'but',
'butter', 'button', 'buy', 'buyer', 'by', 'cabin',
'cabinet', 'cable', 'cake', 'calculate', 'call', 'camera',
'camp', 'campaign', 'campus', 'can', 'cancer', 'candidate',
'cap', 'capability', 'capable', 'capacity', 'capital',
'captain', 'capture', 'car', 'carbon', 'card', 'care',
'career', 'careful', 'carefully', 'carrier', 'carry',
'case', 'cash', 'cast', 'cat', 'catch', 'category',
'cause', 'ceiling', 'celebrate', 'celebration',
'celebrity', 'cell', 'center', 'central', 'century',
'ceremony', 'certain', 'certainly', 'chain', 'chair',
'chairman', 'challenge', 'chamber', 'champion',
'championship', 'chance', 'change', 'changing', 'channel',
'chapter', 'character', 'characteristic', 'characterize',
'charge', 'charity', 'chart', 'chase', 'cheap', 'check',
'cheek', 'cheese', 'chef', 'chemical', 'chest', 'chicken',
'chief', 'child', 'childhood', 'chip', 'chocolate',
'choice', 'cholesterol', 'choose', 'church', 'cigarette',
'circle', 'circumstance', 'cite', 'citizen', 'city',
'civil', 'civilian', 'claim', 'class', 'classic',
'classroom', 'clean', 'clear', 'clearly', 'client',
'climate', 'climb', 'clinic', 'clinical', 'clock', 'close',
'closely', 'closer', 'clothes', 'clothing', 'cloud',
'club', 'clue', 'cluster', 'coach', 'coal', 'coalition',
'coast', 'coat', 'code', 'coffee', 'cognitive', 'cold',
'collapse', 'colleague', 'collect', 'collection',
'collective', 'college', 'colonial', 'color', 'column',
'combination', 'combine', 'come', 'comedy', 'comfort',
'comfortable', 'command', 'commander', 'comment',
'commercial', 'commission', 'commit', 'commitment',
'committee', 'common', 'communicate', 'communication',

ii.

```
while True:
```

```
    print("Modes: words, sentences, numbers, special  
characters")  
    mode=input("Please select a mode: ").lower()  
    if(mode=="words" or mode=="sentences" or  
mode=="numbers" or mode=="special characters"):  
        print()  
        break  
    else:  
        print("Please enter a valid mode.",'\n')  
  
if(mode=="words"):  
    modelist=words  
elif(mode=="sentences"):  
    modelist=sentences  
elif(mode=="numbers"):  
    modelist=numbers  
elif(mode=="special characters"):  
    modelist=specialcharacters  
  
getprompt(mode)
```

```
def getprompt(mode):  
    global prompt  
    while True:  
        numitems=input("Enter the number of "+mode+" you  
would like in the prompt: ")  
        if numitems.isnumeric():  
            break  
        else:  
            print("Please enter a valid number.",'\n')  
    for i in range(int(numitems)):  
        promptlist.append(random.choice(modelist))  
    prompt=' '.join(promptlist)  
    time.sleep(0.5)
```

```
getprompt(mode)  
  
print()  
print("The prompt is:",'\n')  
print(prompt,'\n')  
promptlist=prompt.split(" ")  
time.sleep(0.5)
```