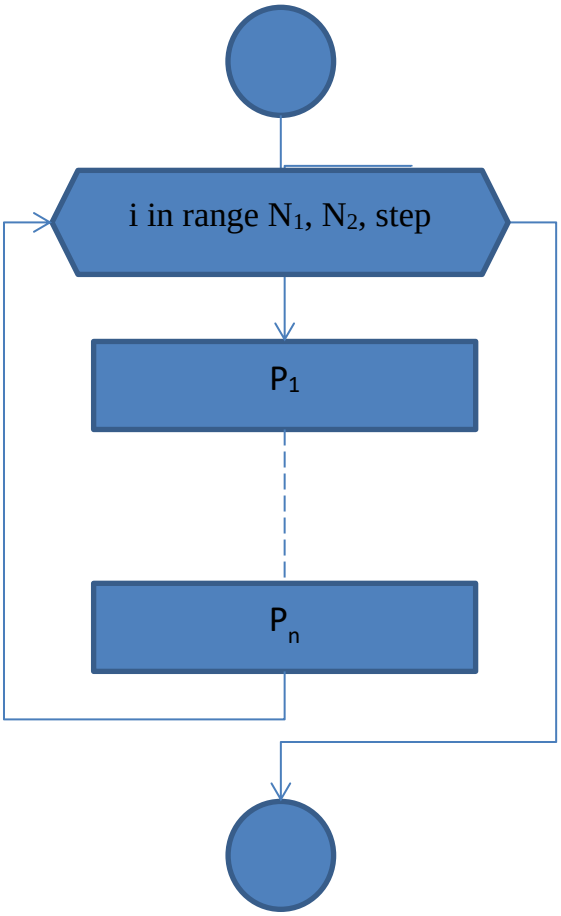
5 ЦИКЛДІК АЛГОРИТМ

5.1 Қарапайым циклдік процесс

Көбінесе бағдарламаларда белгілі бір мәлімдемелерді бірнеше рет орындау керек. Бұл әрекеттер тізбегін қатарынан жиырма-елу рет жазу қисынсыз. Мұндай жағдайларда циклдік есептеулер ұйымдастырылады. Егер белгілі бір қадамдар тізбегі берілген мәнге байланысты бірнеше рет орындалса, **цикл параметрі** деп аталатын,оның **алгоритмі циклдік** деп аталады. Параметр белгілі бір мәнді қабылдаған кезде цикл аяқталады. Белгілі қайталанулар саны бар циклдарды ұйымдастыру үшін Python тілінде **for** операторы қолданылады. Оның алгоритмдегі жалпы көрінісі 42-шісуретте көрсетілген.



Сурет 42 – **For** цикл операторының жалпы түрі

Python тіліндегі **for** циклында әр түрлі жазу формалары болуы мүмкін . Синтаксисын қарастырайық, бірінші түрі. Оны "параметрдің өсіп келе жатқан мәндеріндегі цикл" деп атайық. Егер біз P1...PN параметрлерін циклдың ішінде орындалсын десек онда шегіністерге назар аудару керек.

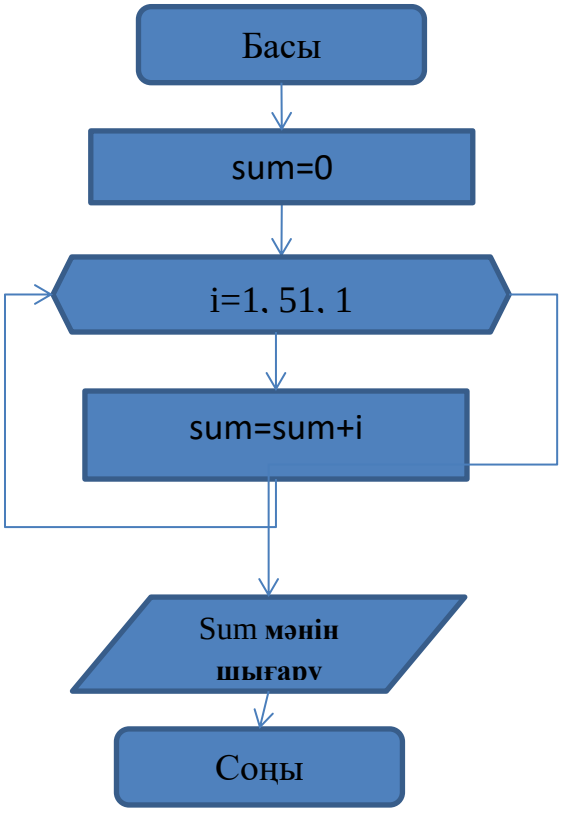
for i **in range** (N_1, N_2, **step**):

P_1
.
.
P_n } Цикл денесі

мұндағы **for** (үшін) – қызмет сөзі; **i** –элементтер мәні сақталатын айнымалы атауы, **P_1,...,P_n** - операторлар; **in** - в; **range** – **Python** тілінің кірістірілген функциясы; **step** - қадамы, міндетті емес параметр.

Range функциясының аргументтері тек бүтін сандар болуы керек. Бұл құрылымның **for** цикл операторының жұмысы келесід. Циклге бірінші рет кірген кезде **i** цикл параметрі N_1 төменгі шекарасының шамасына тең мән алады және цикл денесінде оператор немесе операторлар орындалады. Содан кейін параметр мәні **step** мәніне артады және цикл денесі қайтадан орындалады. Мұндай әрекеттер цикл параметрінің мәні N_2-1 мәніне тең болғанша қайталанады, содан кейін циклден шығу жүзеге асырылады. Егер қадам аргументі **range** функциясында жоқ болса, онда цикл параметрін өзгерту қадамы бірлікке(1) тең болады.

Есеп 5.1. 1-ден 50-ге дейінгі бүтін сандардың қосындысын табыңыз. Есепті шешу алгоритмінің блок-схемасы 43-ші суретте көрсетілген.



Сурет 43 – 5.1 есептің шешу блок-схемасы

Цикл бірінші рет енгізілген кезде цикл параметрі бірлікке тең мәнді қабылдайды және **sum=sum+i** операторы орындалады, содан кейін **i**

параметрі біртіндеп бірлікке тең қадам мәніне көбейтіледі және әр уақытта циклде **sum=sum+i** операторы орындалады. Цикл параметрі **есептегіш** деп аталады, яғни мәні 50 -ге жеткенде цикл тоқтайды. Листингте есептің бағдарлама коды бар:

```
sum=0
for i in range(1, 51, 1):
   sum=sum+i
print("Сумма=", sum)
```

Егер **step** параметрі міндетті емес деп санасаңыз, онда цикл тақырыбын басқаша жазуға болады:

```
sum=0
for i in range(1, 51):
   sum=sum+i
print("Сумма=", sum)
```

Дәл осы бағдарламаны ,цикл параметрдің кему мәні бойынша жазайық. Оны **for** цикл операторының екінші түрі деп алайық.Range функциясындағы соңғы аргумент минус бірге тең. Сәйкес, **i** цикл параметрі 50-ден 1-ге дейін өзгереді, 0 мәні қайта таңдауда қосылмайды:

```
sum=0
for i in range(50, 0, -1):
   sum=sum+i
print("Сумма=", sum)
```

For цикл операторының үшінші формасын жазуда **range** функциясын және жоғарғы шекараның мәнін көрсететін бір параметрді қолдана отырып ұйымдастыруға болады. Осылайша, цикл параметрі 0-ден N_2-1 мәніне дейін өзгереді. Жалпы түрдегі оператордың синтаксисін келесідей жазуға болады:

for i **in range** (N_2):

P_1
.
.
P_n

Тело цикла

«1-ден 50-ге дейінгі бүтін сандардың қосындысын табу» бағдарламасын енді келесі операторлар тізбегі ретінде жазуға болады:

```
sum=0
for i in range(51):
   sum=sum+i
```

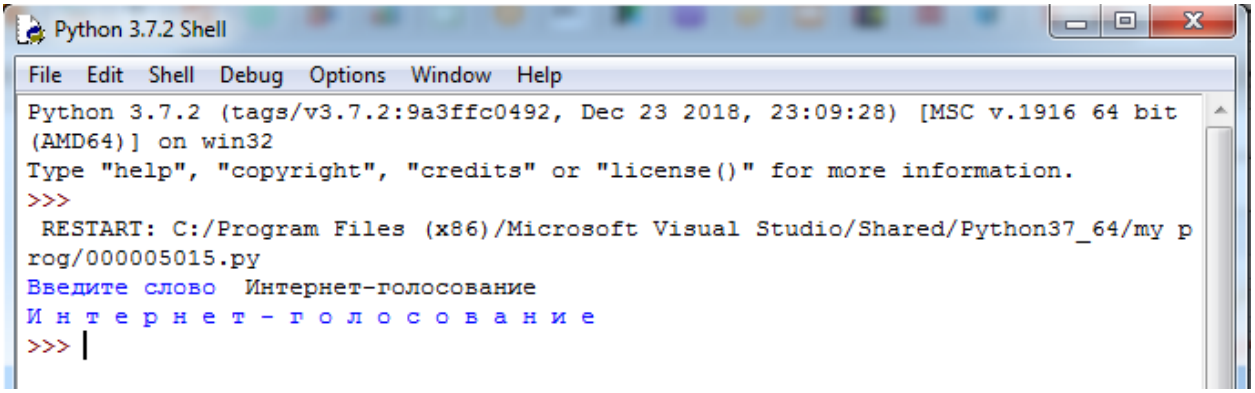
```
print("Сумма=", sum)
```

Қарастырылған барлық бағдарламалардың нәтижесі-1275 санына тең болады.

Біз төменде **for** цикл операторын қолданудың кең мүмкіндіктерін қарастырамыз, енді таңбалар тізбегін өңдеумен байланысты тағы бір мысал келтіреміз:

```
slovo=input("Введите слово  ")
for i in slovo:
    print(i, end=" ")
```

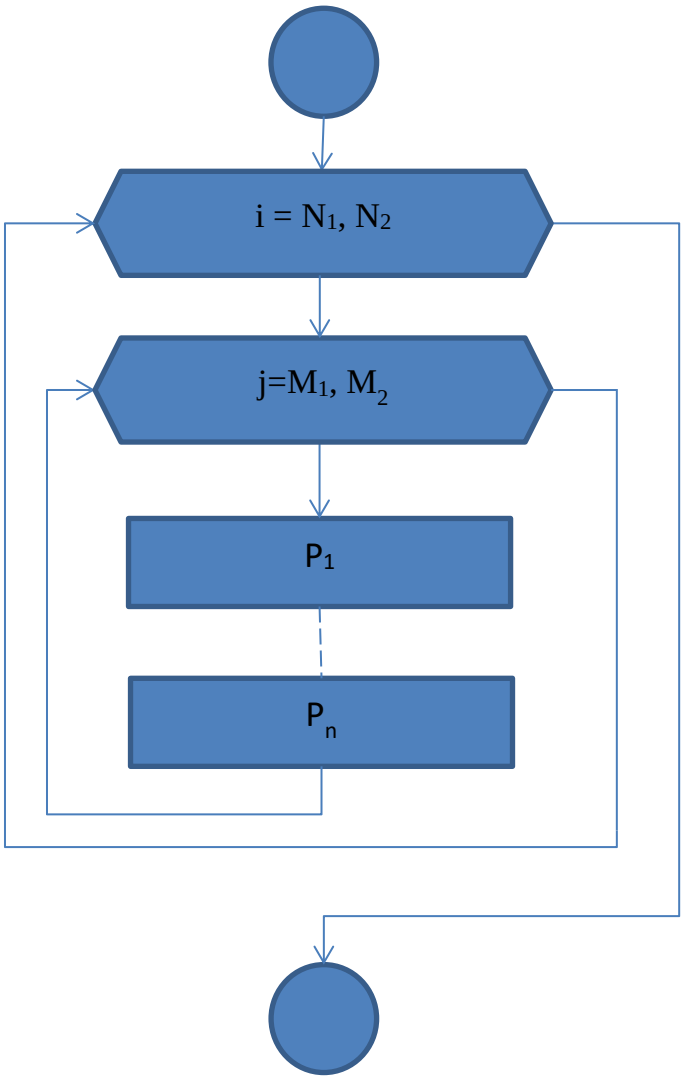
Жоғарыда келтірілген листингтен көрініп тұрғандай, циклде пайдаланушы пернетақтадан енгізетін сөздің(жолдың) таңбалары қайта есептеледі. Бағдарлама жұмысының нәтижесі 44-ші суретте көрсетілген. **Print** операторында, жоғарыда қарастырылған мысалдардан айырмашылығы, кішігірім қосымша қолданылады, атап айтқанда **end**=" " операторы, бұл бағдарлама нәтижелерін бұрын жасалғандай бағанға емес, жолға орналастыруға мүмкіндік берді.



Сурет 44 – **For** циклмен таңбалар тізбегін өңдеуге арналған бағдарлама нәтижесі

5.2 Күрделі циклдік процесс. Кірістірілген циклдар

Егер цикл денесі циклдік құрылым болса, онда мұндай циклдер кірістірілген деп аталады. Басқа циклді қамтитын *цикл* **сыртқы** деп аталады, ал басқа циклдің денесіндегі *цикл* **ішкі** деп аталады. Күрделі циклдік процесс алгоритмінің блок-схемасының фрагментінің жалпы көрінісі 45-ші суретте көрсетілген.



Сурет 45 – Күрделі циклдік процестің блок-схемасының үзіндісі

Күрделі цикл операторларының синтаксисі төменде келтірілген. Ішкі цикл үшін жасалған шегіністерге және онда орындалатын **P_1**... **P_n** операторларына назар аударыңыз:

for i **in range** (N_1, N_2): # сыртқы цикл
  for j **in range** (M_1, M_2): # ішкі цикл
    P_1
    .
    .
    P_n
    } цикл денесі

Күрделі циклдік процестің жұмысын қарастырайық. Циклге бірінші рет кірген кезде сыртқы **i** цикл параметрі **N_1** мәнін алады. Басқару ішкі циклге беріледі, онда **j** циклінің параметрі **M_1**-ге тең мән алады және ішкі циклде жазылған оператор (операторлар) орындалады. Содан кейін **j** ішкі циклінің параметрі бір-біріне артады (егер қадам аргументі **range** функциясында қабылданбаса) және цикл денесі қайтадан орындалады. Содан кейін **i** сыртқы цикл параметрі бірлікке артып, ішкі цикл қайтадан жұмыс істей бастайды,

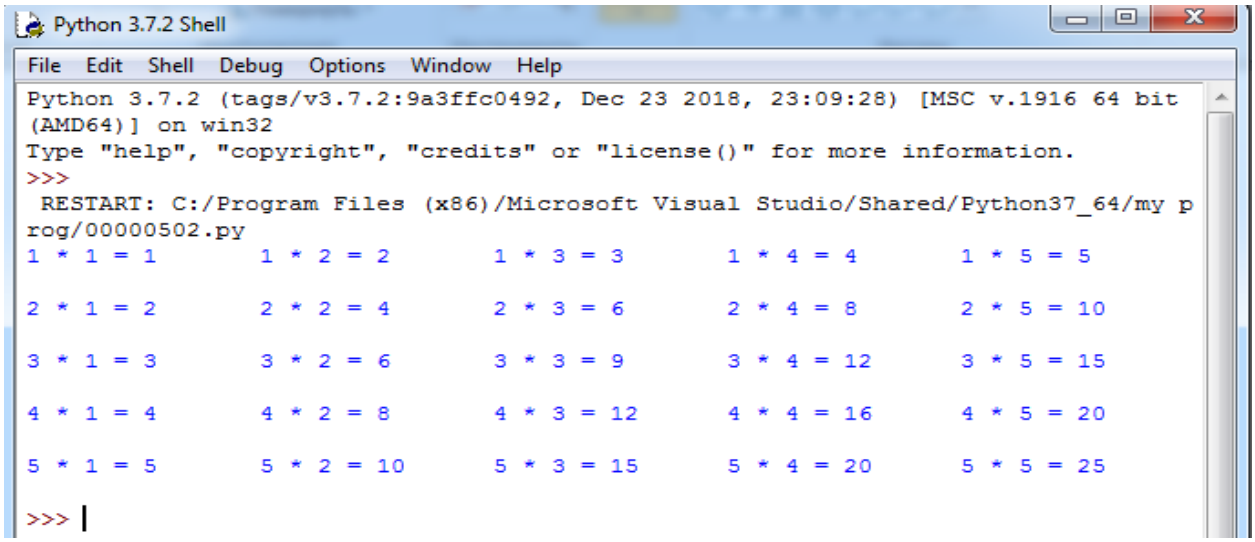
онда **j** цикл параметрі M_1-ден M_2-ге дейін өзгереді және циклдің әр өтуінде $P_1, \ldots, P_n$ операторлары орындалады.

Есеп 5.2. 1-ден 5-ке дейінгі мәндер үшін көбейту кестесін көрсететін қосымшаны жасаңыз.

Шешімі. Шешім тізімі төменде келтірілген:

```
for i in range(1, 6):
    for j in range(1, 6):
        print(i, '*', j, '=', i*j, end='\t')
    print()
```

Сонымен, күрделі циклдік процестің алгоритміне сәйкес ішкі және сыртқы циклдердегі параметрдің өзгеру шекараларын 1-ден 6-ға дейін белгілейміз. Ішкі циклде **print(i, '*', j, '=', i*j, end='\t')** операторы орындалады, онда параметрлер мәндерінің нәтижесі, көбейту және теңдік белгілерін экранға шығару үшін жол өрнектері және іс жүзінде **i*j** әрекеті біріктіріледі, бұл параметрлердің көбеюін қамтамасыз етеді. "Бос" print () операторы (шегініске назар аударыңыз) ішкі циклде орындалмайды, бірақ жауапта шығатын бағандарының арасындағы жолды өткізіп жіберуге қызмет етеді. Бағдарлама жұмысының нәтижесі 46-суретте көрсетілген.



Сурет 46 – көбейту кестесін шығару