

1 Python-ийн хувьсагчид болон өгөгдлийн төрлүүдийг ашиглах

```
x = 5
```

```
y = 3.14
```

```
z = "Python"
```

Дээрх кодоод **x**, **y**, **z** хувьсагчид ямар төрлийн өгөгдөл хадгалагдаж байна вэ?

- a) x - Integer, y - Integer, z - String
- ☒ b) x - Integer, y - Float, z - String
- c) x - Float, y - String, z - Integer
- d) x - String, y - Integer, z - Float

2 Python-д давталт ашиглах

Дараах кодын үр дүн юу вэ?

```
for i in range(3):
```

```
    print(i)
```

- a) 1 2 3
- ☒ b) 0 1 2
- c) 0 1 2 3
- d) Алдаа үүсгэнэ

3 Python-ийн мөрийн төгсгөл ба кодын зохион байгуулалт

Python-д мөрийн төгсгөлд ямар тэмдэг ашиглах шаардлагагүй вэ?

- a) :
- b) ;
- ☒ c) #
- d) ""

4 Хувьсагчдын тодорхойлолт

Дараах кодын үр дүн юу вэ?

```
name = "Alice"
```

```
age = 25
```

```
print(name, age)
```

- ☒ a) Alice 25
- b) name age
- c) "Alice" "25"
- d) Алдаа үүсгэнэ

5 Функц гэж юу вэ?

Python функцийн давуу талуудын нэг нь юу вэ?

- a) Кодыг дахин ашиглах боломжгүй болгоно
- b) Кодыг илүү төвөгтэй болгодог
- ☒ c) Кодыг дахин ашиглах, засварлахад хялбар болгоно
- d) Функц нь заавал аргумент шаарддаг

6 Python функц тодорхойлох

Доорх функцийн үр дүн юу вэ?

```
def greet():
```

```
    print("Сайн байна уу!")
```

```
greet()
```

- ☒ a) Сайн байна уу!
- b) Алдаа гарна
- c) greet() гэж хэвлэнэ
- d) Ямар ч үр дүн гаргахгүй

7 Өгөгдлийн төрлүүдийн ажиллагаа

Доорх кодын үр дүн юу вэ?

```
python
```

```
Copy code
```

```
x = "10"
```

```
y = 5
```

```
print(x + y)
```

- a) 15
- b) 105
- ☒ c) Алдаа гарна
- d) "10 5" гэж хэвлэнэ

7, Алгоритм гэж юу вэ?

Алгоритм гэдэг нь тодорхой асуудлыг шийдвэрлэх, зорилгод хүрэхэд чиглэсэн үйлдлүүдийн дараалал юм. Дараахын аль нь алгоритмын шинж чанар биш вэ?

- a) Тодорхой байдал
- b) Эцсийн байдал
- ☒ c) Дарааллын чөлөөт байдал
- d) Үр ашигтай байдал

8, Псевдокод нь алгоритмыг дүрслэхэд ашигладаг. Псевдокодын давуу талуудын нэг нь юу вэ?

- ☒ a) Програмчлалын хэлний нарийн мэдлэг шаарддаг
- b) Программистуудад л ойлгомжтой
- c) Ойлгомжтой байдал, бичихэд хялбар
- d) Код бичих процессод алдаа гарах магадлал их

9, Алгоритмын гүйцэтгэл

Алгоритмын гүйцэтгэл нь юу вэ?

- ☒ a) Алгоритм нь өгөгдлийг боловсруулах алхамуудын дараалал
- b) Алгоритм нь зөвхөн логик үйлдлүүдийг гүйцэтгэдэг
- c) Алгоритм нь оролтын хэмжээний хамааралгүй
- d) Алгоритм нь нэг удаа гүйцэтгэгддэг

10, Доорх Big O тэмдэглэгээгээс аль нь тогтмол хугацаа ($O(1)$) илэрхийлж байна вэ?

- a) $O(n)$
- b) $O(\log n)$
- c) $O(n^2)$
- ☒ d) $O(1)$

11, Эрэмбийн алгоритм гэж юу вэ?

Эрэмбийн алгоритм нь:

A) Оролтын өгөгдлийг илүү сайн зохион байгуулах зорилготой алгоритм.

☒ B) Өгөгдлийг тодорхой дарааллаар (өсөх, буурах гэх мэт) эрэмбэлэх зорилгоор ашиглагддаг.

C) Ямар нэгэн тооны математик үйлдлийг гүйцэтгэх алгоритм.

D) Програмын гүйцэтгэлийг үнэлэх арга.

12, Bubble Sort алгоритмын үйлдлийн тоо $O(n^2)$ гэсэн утгатай юу?

$O(n^2)$ нь:

A) Алгоритм нь оролтын хэмжээтэй шууд пропорционал.

☒ B) Алгоритмын үйлдлийн тоо нь оролтын хэмжээний квадраттай хамааралтай.

C) Алгоритм нь тогтмол хугацаанд гүйцэтгэгддэг.

D) Алгоритм нь оролтын хэмжээг нэмэгдүүлэхэд огт өөрөөр гүйцэтгэгддэг.

13, Массив болон холбоослогдсон жагсаалтын (Linked List) хоорондох ялгааг тайлбарла.

(Тайлбарт: өгөгдлийг хадгалах арга, санах ой, мэдээллийг боловсруулах хурд гэх мэт)

Arrays нь элементүүдийг санах байрлалд хадгалдаг

14, Стек (Stack) болон мод (Tree) мэдээллийн бүтцийн ялгааг дурдаж, аливаа нэг жишээг ашиглан тэдгээрийн үйл ажиллагааг тайлбарла.

(Жишээнд: хэрхэн ажилладаг, хэрэглээ, давуу болон сул талууд)

15, AVL модны тэнцвэрийн хүчин зүйл (Balance Factor) гэж юу вэ?

A. Зангилааны баруун салбарын өндөр

B. Зангилааны зүүн салбарын өндөр

☒ C. Зангилааны зүүн болон баруун салбарын өндрийн зөрүү

D. Зангилааны нийт тоо

16, AVL мод болон энгийн хоёртын хайлтын модны ялгаа юу вэ?

☒ A. AVL мод нь всегда тэнцвэртэй, хоёртын хайлтын мод нь тэнцвэргүй байж болно.

B. AVL мод нь зөвхөн оруулалтын үйлдлүүдийг дэмждэг.

C. AVL мод нь хайлт хийхэд удаан, хоёртын хайлтын мод нь хурдан.

D. AVL мод нь давхаргын мод биш.

17, Stack (LIFO) бүтэц нь ямар зарчмаар ажилладаг вэ?

A. Эхний орж ирсэн нь анхандаа гарах

☒ B. Сүүлийн орж ирсэн нь анхандаа гарах

C. Элементийг случайгаар устгах

D. Элементийг жагсаалтаар хадгалах

18, Queue (FIFO) бүтэцийн үндсэн үйлдлүүд юу вэ?

A. push болон pop

B. add болон remove

☒ C. enqueue болон dequeue

D. insert болон delete

19, Графын тухай юу вэ?

A. Граф нь зөвхөн нэг орон болон олон ирмэгээс бүрдсэн өгөгдлийн бүтэц юм.

B. Граф нь олон зангилаа болон тэдгээрийн хоорондын холбоосыг дүрслэн харуулахад ашиглагддаг.

C. Граф нь өгөгдлийг хадгалахын тулд заавал массив байх ёстой.

D. Граф нь зөвхөн тоон өгөгдлийг боловсруулахад хэрэглэгддэг.

20, Чиглэлтэй графын онцлог нь юу вэ?

A. Ирмэгүүдийн чиглэл тодорхой бус байна.

B. Ирмэгүүд нь зөвхөн нэг чиглэлтэй байх бөгөөд А оройноос В орой руу шилжих боломжтой.

C. Ирмэгүүд нь бүх чиглэлд үйлчилдэг.

D. Чиглэлтэй граф нь зөвхөн цэгүүдийн нийлэмжээс бүрддэг.

21, Графыг ямар салбарт ашиглаж болох вэ?

A. Файлын систем доторх өгөгдлийг хадгалах.

B. Нийгмийн сүлжээ болон замын сүлжээний хоорондын харилцааг илэрхийлэх.

C. Дата базад өгөгдлийг шүүх.

D. График интерфэйсийн элементүүдийг боловсруулах.

22, Чиглэлгүй графын онцлог нь юу вэ? (Зөв хариултыг сонгоно уу)

A. Зангилаа хоорондын холболтод чиглэл байхгүй.

B. Зангилаа хоорондын холболтод зөвхөн нэг чиглэл байх ёстой.

C. Ирмэгүүд нь заавал утаснаас холбогдсон байх ёстой.

D. Чиглэлгүй граф нь зөвхөн тоон өгөгдөлд хэрэглэгддэг.

1. Грэйди алгоритм – Минимум зоосны тоо

Даалгавар 1:

Грэйди алгоритмыг ашиглан мөнгөний дүнг хамгийн бага зоосоор төлөх Python програм бич.

Даалгавар:

Таны зорилго бол тодорхой мөнгөн дүнг (жишээлбэл, 83 төгрөг) хамгийн бага зоосоор төлөх явдал юм.

Зоосны төрлүүд нь: 1 төгрөг, 5 төгрөг, 10 төгрөг, 25 төгрөг.

Жишээ үр дүн:

Input: 83 төгрөг

Output: 3x25 төгрөг, 1x5 төгрөг, 3x1 төгрөг

Шаардлага:

Python програм нь

`greedy_coin_change(amount)`

гэсэн функцтэй байх ба дүнг оруулахад зоосны хуваарилалтыг хамгийн бага тоогоор буцаана.

Асуулт 1: Грэйди алгоритмын дагуу хамгийн бага зоосны тоог хэрхэн сонгох вэ?

Та 67 төгрөгтэй бол дараах зоосуудыг ашиглан хамгийн бага

тооны зоосыг сонгож дүгнэнэ үү: 1, 5, 10, 25.

- a) 4 зоос (2x25, 1x10, 2x1)
- b) 5 зоос (1x50, 1x10, 1x5, 2x1)
- ☒ c) 6 зоос (2x25, 1x5, 2x1)
- d) 3 зоос (1x50, 1x10, 1x5)

Асуулт 2: Аль алгоритм нь Грэйди алгоритмын зарчмыг ашиглан шийдэгддэг вэ?

- a) Quick Sort
- ☒ b) Kruskal's Algorithm
- c) Merge Sort
- d) Depth-First Search

2. Хаффман кодчилал

Даалгавар 2:

Тэмдэгтүүдийн давтамж дээр үндэслэн Хаффман код үүсгэх Python програм бич. Програм нь тэмдэгтүүдийг шахаж, хамгийн бага битээр хэрхэн илэрхийлэхийг харуулна.

Даалгавар:

Өгөгдсөн тэмдэгтүүдийн давтамжийг ашиглан Хаффман мод үүсгэнэ.

Тэмдэгт бүрийн Хаффман кодыг олох.

Жишээ өгөгдөл:

python

Copy code

```
Тэмдэгтүүд: {'A': 5, 'B': 9, 'C': 12, 'D': 13, 'E': 16, 'F': 45}
```

Жишээ үр дүн:

F: 0

C: 100

D: 101

A: 1100

B: 1101

E: 111

Шаардлага:

Python програм нь дараах байдлаар ажиллана:

```
huffman_code(frequencies)
```

функц ашиглан тэмдэгтүүдийн давтамжийг оруулж, тэдгээрийн Хаффман кодыг гаргана.

3. Bubble Sort болон Insertion Sort Хэрэгжүүлэлт ба Харьцуулалт

Даалгавар 3:

Python хэл дээр **Bubble Sort** болон **Insertion Sort** алгоритмуудыг хэрэгжүүлээд, өгөгдсөн жагсаалтыг эрэмбэлнэ үү.

Жишээ:

python

Copy code

```
Input: [64, 34, 25, 12, 22, 11, 90]
```

`bubble_sort()` болон `insertion_sort()` функцийг бичээд, эрэмбэлсэн жагсаалтыг буцаана.

Хоёр алгоритмын гүйцэтгэлийг цаг хугацааны хувьд харьцуулж, үр дүнг тайлбарлана.

Жишээ үр дүн:

python

Copy code

```
Bubble Sort үр дүн: [11, 12, 22, 25, 34, 64, 90]
```


Insertion Sort үр дүн: [11, 12, 22, 25, 34, 64, 90]

Bubble Sort хугацаа: 0.002 секунд

Insertion Sort хугацаа: 0.001 секунд

3. Алгоритмын гүйцэтгэлийн шинжилгээ

Асуулт 1: $O(n)$ гэж юуг илэрхийлдэг вэ?

- ☒ a) Алгоритмын орон зайн хэрэглээг
- b) Алгоритмын хамгийн их боломжит гүйцэтгэлийг
- c) Алгоритмын хамгийн бага гүйцэтгэл
- d) Алгоритмын тогтмол гүйцэтгэлийг

4. Бинар мод үүсгэх ба хайлтын мод

Даалгавар 4:

Python дээр бинар хайлтын мод (Binary Search Tree) үүсгэж, дараах үйлдлүүдийг гүйцэтгэнэ үү:

Уг модонд утгуудыг нэмэх (insert)

Утга хайх (search)

Хамгийн бага болон хамгийн их утгыг олох (find_min, find_max)

Жишээ:

Input: [20, 9, 25, 5, 12, 15, 30]

Гүйцэтгэсэн үйлдлүүд:

insert() функц ашиглан модонд утгуудыг нэмнэ.

search() функц ашиглан тодорхой утга байгаа эсэхийг шалгана.

find_min() болон find_max() функц ашиглан хамгийн бага болон хамгийн их утгыг олох.

Жишээ үр дүн:

Minimum value: 5

Maximum value: 30

Search 12: Found

Search 18: Not found

4. Функц ба буцаах утга

Асуулт 1: Дараах функц ямар утга буцаах вэ?

```
def test_func(x):  
    if x % 2 == 0:  
        return x * 2  
    else:  
        return x + 2  
  
print(test_func(5))
```

- a) 10
- b) 7
- c) 12
- d) 5