

កិច្ចការផ្ទះ (Homework 01)

ប្រធានបទ ៖ “ ការកំណត់កំហុស Python និងតស៊ូតាងសម្រាប់ការបង្កើត Expert System ”

១.HW1.1 -សេចក្តីពន្យល់ជាភាសាខ្មែរ តើ Expert System ខុសពីកម្មវិធីរកពាក្យគន្លឹះធម្មតាយ៉ាងដូចម្តេច ?
ចូរលើកឧទាហរណ៍ពីការរៀន Python មកបញ្ជាក់ឱ្យបានមួយយ៉ាងតិចតាមប្រភពផ្លូវការ។

ចម្លើយ ៖

មុននឹងឈានដល់ការពន្យល់យើងត្រូវយល់ដឹងអំពី កម្មវិធីរកពាក្យគន្លឹះ និង Expert System ជាមុនសិន។
_កម្មវិធីរកពាក្យគន្លឹះ ឬក៏យើងអាចហៅបានថាជា (Keyword Search) គឺជាការស្វែងរកទិន្នន័យដែលដូចគ្នា
បេះបិទ ហើយវាមិនយល់ន័យ ឬអាចសម្រេចចិត្តបានឡើយ។ ឧទាហរណ៍ថា យើងចូលទៅ Google ដើម្បី
Search រកអ្វីមួយ។ ឥឡូវគិតថាអ្នករកមើលពាក្យថា Learn នោះវានឹងបង្ហាញប្រធានបទផ្សេងដូចជា Learn
code, Learning, Learn English ជាដើម។ ទាំងនេះមានន័យថាវាចាប់យកតែឯកសារណាដែលមានពាក្យ
Learn ប៉ុណ្ណោះ។

ឧទាហរណ៍ដោយប្រើកូដក្នុង Python៖

```
code = """
score = 65
if score >=50:
    print("Pass")
```

```
if score >=80:
    print("Excellent")
"""
```

```
keyword = "if"
```

```
for line in code.splitlines():
    if keyword in line:
        print(line)
```

លទ្ធផលដែលចេញមក

```
if score >=50:
if score >=80:
```

>បកស្រាយកូដ

. code គឺជាអត្ថបទកូដ Python ដែលយើងចង់ស្វែងរក។
.keyword = "if" គឺជាពាក្យដែលយើងចង់រក។
.splitlines() បំបែកអត្ថបទជាបន្ទាត់។ [1]។
.if keyword in line ពិនិត្យថា បន្ទាត់នោះមានពាក្យ if ឬអត់។
.print(line) បង្ហាញបន្ទាត់ដែលមានពាក្យនោះ។

ដំណើរការកូដប្រព្រឹត្តទៅដោយ

១. ដំបូងឡើយកូដបានដំណើរការដោយប្រើពាក្យដែលត្រូវរកគឺ if ដែលយើងបានតាំងនៅក្នុងកូដជា Keyword។
ហើយ

បន្ទាប់មកនៅពេលដែលវាបានរកឃើញ វានឹងចាប់យកតែជួរណាដែលមានពាក្យ if ហើយកាត់ផ្តាច់ជួរ
នោះចេញពីគេមកឈរតែម្នាក់ឯងដោយសារ មុខងារ splitlines() [1]។ ក្រោយមកយើងក៏ទទួលបានលទ្ធផល
ដូចដែល

បានបង្ហាញខាងលើ។

ចំណុចសំខាន់ដែលយើងគួរដឹងត្រង់ចំណុចនេះគឺ៖

កម្មវិធីបានរកឃើញ if ប៉ុន្តែវាមិនបានយល់ថា តើវាមានគោលបំណងអ្វីឬបង្កើតឡើងសម្រាប់អ្វីឡើយ។

_ Expert System (ឬប្រព័ន្ធអ្នកជំនាញ) គឺជាប្រព័ន្ធ AI ដែលមាន **ចំណេះដឹង (knowledge base)** ជាច្បាប់ IF–
THEN ដែលមនុស្សជាអ្នកជំនាញបានកំណត់ និង **ម៉ាស៊ីនអនុមាត (inference engine)** ដែលយកច្បាប់ទាំងនោះ
មកអនុវត្តលើករណីជាក់លាក់ ដើម្បីផ្តល់ការសន្និដ្ឋាន [2]។ ប្រព័ន្ធ AI នេះត្រូវបានគេបង្កើតឡើងដោយផ្អែកលើគោល
ការណ៍សំខាន់ៗមួយចំនួន ដែលរួមមាន ចំណេះដឹងពិតប្រាកដ (Knowledge Base), ការពិតជាក់ស្តែង (Facts), ត
ស្តុតាង ច្បាប់ និងលក្ខខណ្ឌពិតប្រាកដ ដើម្បីធ្វើការសន្និដ្ឋាន (Inference Engine) [2] ហើយចុងក្រោយគឺសេចក្តី
ពន្យល់

(Explanation Facility) ។

ឧទាហរណ៍ទី ១៖

ស្រមៃមើលថាយើងមានអាការៈដូចជា៖

.ក្តៅ

.ក្អក

.ឈឺក្បាល

ហើយនៅពេលដែលយើងបញ្ចូលទៅប្រព័ន្ធ វានឹងអនុវត្តតាមច្បាប់ដែលគេកំណត់ថា

IF fever = YES

AND cough = YES

AND headache = YES

THEN possible flu

កន្លែងនេះមានន័យថា៖

បើសិនជាយើងក្តៅខ្លួន ឈឺក្បាល និងក្អក នោះវានឹងបង្កើតការសន្និដ្ឋានថាយើងអាចនឹងមានជំងឺផ្តាសាយ។

ប៉ុន្តែចូរចាំថា គ្រាន់តែដើរតាមច្បាប់ដែលគេកំណត់មក វាមិនមែនជាការវិនិច្ឆ័យពិតប្រាកដទេ។

ដូច្នេះយើងនឹងសន្និដ្ឋានអំពីតាមខុសគ្នារវាង ការស្វែងរកពាក្យគន្លឹះ និង Expert System

ចំណុចប្រៀបធៀប	Keyword Search	Expert System
និយមន័យ	កម្មវិធីស្វែងរកពាក្យដែលដូចគ្នាក្នុងអត្ថបទឬទិន្នន័យ	ប្រព័ន្ធ AI ដែលប្រើចំណេះដឹង និងច្បាប់ដើម្បីដោះស្រាយបញ្ហា
របៀបធ្វើការ	ស្វែងរកពាក្យដែលត្រូវគ្នា	វិភាគព័ត៌មានដោយប្រើច្បាប់ IF-THEN
ការយល់ដឹង	មិនយល់អត្ថន័យនៃព័ត៌មានទេ	ប្រើចំណេះដឹងដែលបានកំណត់ដើម្បីសន្និដ្ឋាន
ការសម្រេចចិត្ត	មិនធ្វើការសម្រេចចិត្តទេ	អាចផ្តល់ការសន្និដ្ឋាន ឬជំនួយ
ធាតុសំខាន់	ពាក្យគន្លឹះ (Keyword)	Knowledge Base និង Inference Engine
ឧទាហរណ៍	ស្វែងរកពាក្យ if ក្នុងឯកសារ Python	ប្រព័ន្ធជួយពិនិត្យជំងឺអ្នកជំងឺ
លទ្ធផល	បង្ហាញបង្អាត ឬអត្ថបទដែលមានពាក្យ	ផ្តល់លទ្ធផលវិភាគ ឬជំនួយ

២. HW1.2 - បង្កើតករណីខ្លីៗ ៣ ដោយសម្រាប់ករណីនីមួយៗ បង្ហាញ Code Snippet, Evidence, Expected/Actual Result ប្រសិនបើពាក់ព័ន្ធ, Error Type, Cause និង Hint 1 ។ កុំយកករណីដែលមិនអាចពន្យល់ដោយស្មុគស្មាញ។

ក. Syntax Error

ខ. Runtime Error

គ. Logic Error

ចម្លើយ៖

ក. Syntax Error

Field	បំពេញ
Case ID	ក. PY-001
Exercise / Specification	បង្ហាញថា Pass ប្រសិនបើពិន្ទុ ≥ 50 (លក្ខខណ្ឌ: score ≥ 50)
Code Snippet / Input	python score = 65 if score >= 50 print('Pass')
Evidence	SyntaxError: expected ':'
Expected / Actual Result	Expected: Pass · Actual: ក្នុងមិនដំណើរការសោះ (error មុនដំណើរការ)
Error Type	Syntax Error
Cause	បង្គាត់ if ខ្វះ colon : នៅចុងលក្ខខណ្ឌ
Hint 1	បន្ថែម : នៅចុងបង្គាត់ if

Reference / Verified By	Python Docs – Compound statements: https://docs.python.org/3/reference/compound_stmts.html (ពិនិត្យដោយការរត់កូដ) [3], [4]
--------------------------------	--

ខ. Runtime Error

Field	បំពេញ
Case ID	ខ. PY-002
Exercise / Specification	បញ្ហា Pass ប្រសិនបើពិន្ទុ ≥ 50 (ពិន្ទុទទួលបានពីអ្នកប្រើ)
Code Snippet / Input	python score = input("Enter score: ") # ឧទាហរណ៍បញ្ចូល 65 if score >= 50: print('Pass')
Evidence	TypeError: '>=' not supported between instances of 'str' and 'int'
Expected / Actual Result	Expected: Pass · Actual: កម្មវិធីឈប់នៅបន្ទាត់ if
Error Type	Runtime Error
Cause	input() ត្រឡប់ជា string ("65") ប៉ុន្តែលក្ខខណ្ឌប្រៀបធៀបជាមួយលេខ 50
Hint 1	បម្លែងតម្លៃទៅជាលេខជាមុនសិន ដោយប្រើ int(score)
Reference / Verified By	Python Docs – Built-in Functions (input, int): https://docs.python.org/3/library/functions.html (ពិនិត្យដោយការរត់កូដ) [3], [5]

គ. Logic Error

Field	បំពេញ
Case ID	គ. PY-003

Exercise / Specification	បង្ហាញ Pass ប្រសិនបើពិន្ទុ ≥ 50 (ពិន្ទុ 50 ត្រូវតែជាប់)
Code Snippet / Input	python score = 50 if score > 50: print('Pass') else: print('Fail')
Evidence	Expected output: Pass · Actual output: Fail (គ្មាន error message)
Expected / Actual Result	Expected: Pass · Actual: Fail
Error Type	Logic Error
Cause	ប្រើ > ជំនួស >= ដូច្នេះពិន្ទុ 50 មិនត្រូវបានរាប់ថាជាប់
Hint 1	ពិនិត្យតម្លៃព្រំដែន (boundary): តើ 50 គួរតែ Pass ឬ Fail?
Reference / Verified By	Python Docs – Comparisons: https://docs.python.org/3/reference/expressions.html#comparisons (ពិនិត្យដោយការរត់កូដ) [6]

៣. HW1.3 -សរសេរចម្លើយ៖

មូលហេតុចម្រើនបំផុតដែល Logic Error ត្រូវការលក្ខខណ្ឌសំខាន់គឺដើម្បីប្រៀបធៀបលទ្ធផលដែលកម្មវិធីផ្តល់ឱ្យជាមួយលទ្ធផលដែលគួរតែទទួលបាន។ វាមិនបានបង្ហាញយើងថាកំហុសបានកើតឡើងទេ តែលទ្ធផលដែលទទួលបានខុសពីការរំពឹងទុក ដូច្នេះជាដាច់ខាតយើងត្រូវតែមានលក្ខខណ្ឌច្បាស់លាស់។

— ដូច្នេះប្រព័ន្ធគួរតែឆ្លើយតបទៅនឹងករណីដែលត្រូវបានគ្រប់គ្រាន់ គឺនៅពេលដែលព័ត៌មានដែលអ្នកប្រើប្រាស់ ឬយើងបានផ្តល់ទៅឱ្យប្រព័ន្ធនោះ មិនទាន់គ្រប់គ្រាន់ដើម្បីឱ្យវាសម្រេចចិត្ត។

សំណួរសំខាន់ដែលប្រព័ន្ធត្រូវការសម្រាប់កំណត់ថា៖

.តើកម្មវិធីរបស់យើងត្រូវធ្វើអ្វី?

.តើលទ្ធផលដែលចេញមកនោះមានលក្ខណៈយ៉ាងដូចម្តេច?

.តើលទ្ធផលចេញមកខុសពីអ្វីដែលយើងចង់បានឬអត់?

តទ្បូវយើងលើកឧទាហរណ៍មួយ៖

បើតាមជាង ពេលកុំព្យូទ័របើកមិនដំណើរការ ហើយឆ្នាំងសាកច្របល់រាជោគហើយភ្លើងមិនចូល ដូច្នេះគេសន្និដ្ឋានថាកុំ

ព្យុទ្ធនោះខូចឆ្នាំងសាកថ្ម។

តែពេលយើងសួរទៅអតិថិជន គាត់ប្រាប់ថា កុំព្យូទ័របើកមិនដំណើរការ ហើយរឿងឆ្នាំងសាកនោះគាត់មិនច្បាស់ថា សាកចូលឬអត់ដែរ។

ដូច្នេះត្រង់ចំណុចនេះ ទោះបីជា System ឬជាដក់ដោយ ក៏គេមិនអាចសម្រេចបានដែរថាកុំព្យូទ័រខូច ឬឆ្នាំងសាកខូច។ ដូច្នេះចុងក្រោយ យើងសន្និដ្ឋានបានថា តស៊ូតាងមិនទាន់គ្រប់គ្រាន់សម្រាប់ធ្វើការសន្និដ្ឋាន។

ឆ្ងុយទៅវិញ យើងអាចសួរបន្ថែមថាឆ្នាំងសាកថ្មនោះដំណើរការឬអត់?

សេចក្តីប្រកាសអំពីការប្រើប្រាស់ AI (AI Usage Disclosure)

ក្នុងការធ្វើកិច្ចការផ្ទះនេះ ខ្ញុំបានប្រើជំនួយពី AI ដូចខាងក្រោម៖

- [ChatGPT / Microsoft Copilot] ត្រូវបានប្រើដើម្បីជួយឱ្យខ្ញុំយល់អំពីគំនិតនៃ Expert System, Keyword Search និងប្រភេទកំហុសក្នុង Python ព្រមទាំងជួយកែលម្អការពន្យល់ឱ្យកាន់តែច្បាស់។
- Claude (Anthropic) ក្នុង Microsoft Word ត្រូវបានប្រើដើម្បីជួយកែអក្ខរាវិរុទ្ធ ធរណីមាត្រ និងទម្រង់អត្ថបទ ព្រមទាំងរៀបចំឯកសារយោង និងការដកស្រង់ក្នុងអត្ថបទតាមរចនាប័ទ្ម IEEE។

ខ្ញុំបានពិនិត្យខ្លឹមសារ កូដ និងប្រភពឯកសារយោងទាំងអស់ដោយខ្លួនឯង ហើយគំនិត និងការពន្យល់សំខាន់ៗនៅក្នុង កិច្ចការនេះ គឺជារបស់ខ្ញុំផ្ទាល់។

Samath pannharoth

ឯកសារយោង (References)

- [1] Python Software Foundation, “Built-in Types: str.splitlines(),” *The Python Standard Library*, Python 3 Documentation. Accessed: Oct. 23, 2026. [Online]. Available: <https://docs.python.org/3/library/stdtypes.html#str.splitlines>
- [2] Encyclopædia Britannica, “Inference engine,” *Encyclopædia Britannica*. Accessed: Oct. 23, 2026. [Online]. Available: <https://www.britannica.com/technology/inference-engine>

- [3] Python Software Foundation, “Errors and Exceptions,” *The Python Tutorial*, Python 3 Documentation. Accessed: Oct. 23, 2026. [Online]. Available: <https://docs.python.org/3/tutorial/errors.html>
- [4] Python Software Foundation, “Compound statements,” *The Python Language Reference*, Python 3 Documentation. Accessed: Oct. 23, 2026. [Online]. Available: https://docs.python.org/3/reference/compound_stmts.html
- [5] Python Software Foundation, “Built-in Functions,” *The Python Standard Library*, Python 3 Documentation. Accessed: Oct. 23, 2026. [Online]. Available: <https://docs.python.org/3/library/functions.html>
- [6] Python Software Foundation, “Expressions: Comparisons,” *The Python Language Reference*, Python 3 Documentation. Accessed: Oct. 23, 2026. [Online]. Available: <https://docs.python.org/3/reference/expressions.html#comparisons>