

Student ID: .....



**EXAMINATIONS — 2011**  
**MID-YEAR**

**NWEN 241**  
**SYSTEMS PROGRAMMING**

**Time allowed:** THREE HOURS

**Instructions:** The examination contains 6 questions. You must answer ALL questions

Each question is worth 30 marks.

The exam consists of 180 marks in total.

Non-programmable calculators are allowed.

Paper foreign to English language dictionaries are allowed.

Electronic dictionaries and programmable calculators are not allowed.

Student ID: .....

**Question 1 - Python fundamentals**

**[30 marks]**

a) [5 marks] Some of following strings are legal Python identifiers and some are not. For each one state if the name is legal or not and if not explain why:

- 1) lambda
- 2) 4squared
- 3) days\_of\_week3
- 4) square-root
- 5) daysOfWeek

b) [5 marks] Consider the following code fragment:

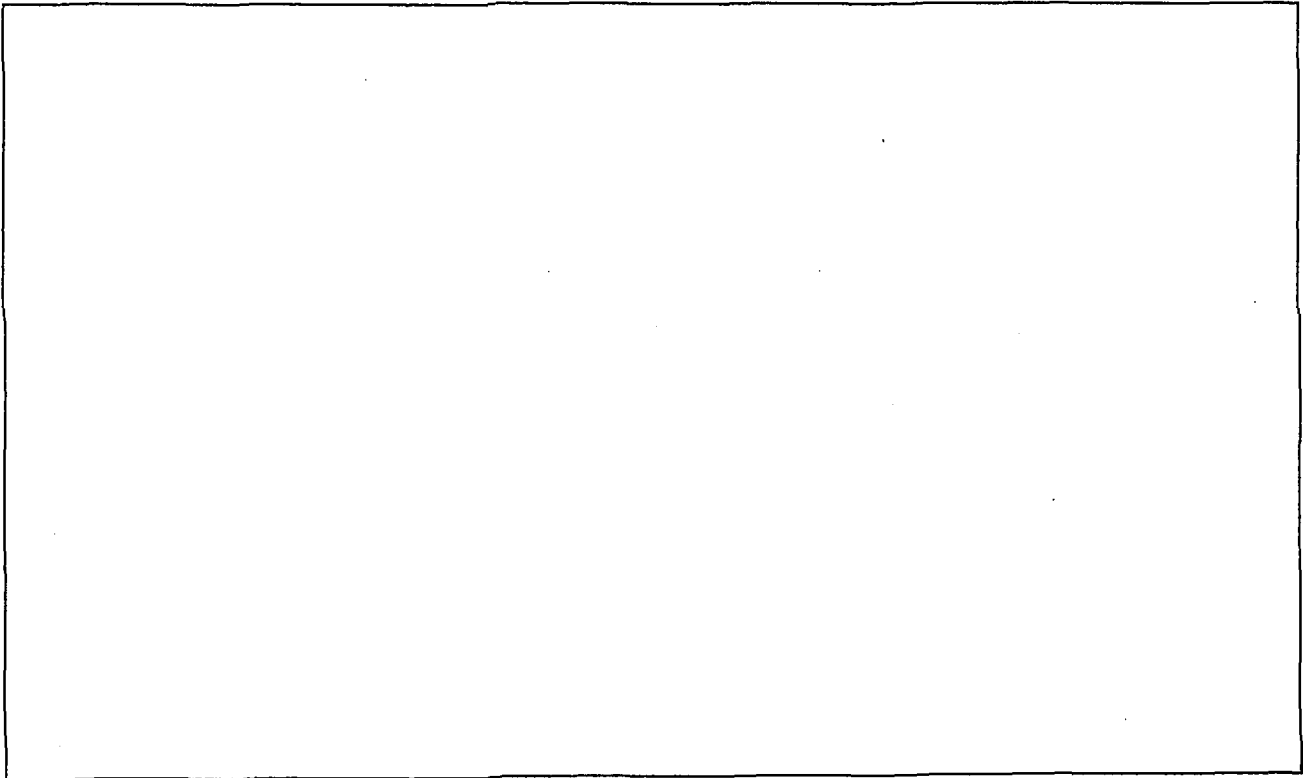
```
>>> greeting = 'Hello, world!'
>>> greeting[0] = 'J'
TypeError: object does not support item assignment
```

Explain why this error message occurs.

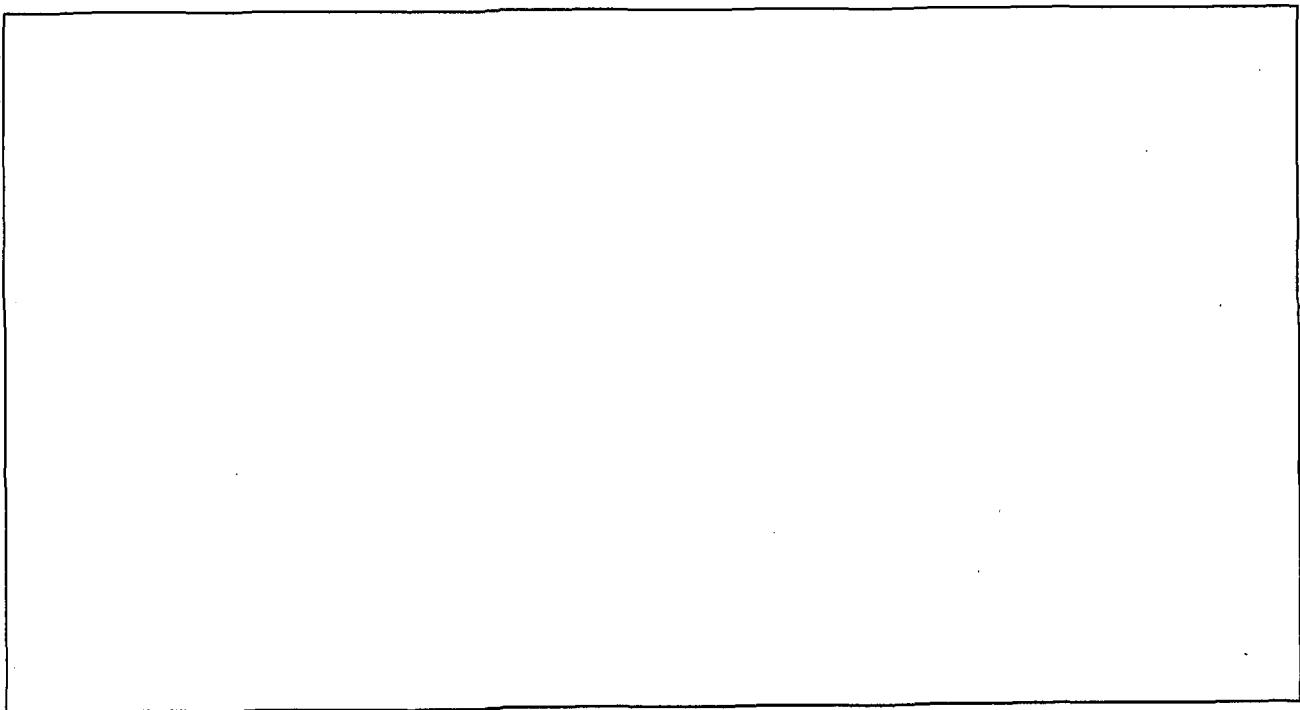
Student ID: .....

c) [20 marks] For each of the following Python keywords briefly explain what it does and write a short piece of code to illustrate its use.

1) Try

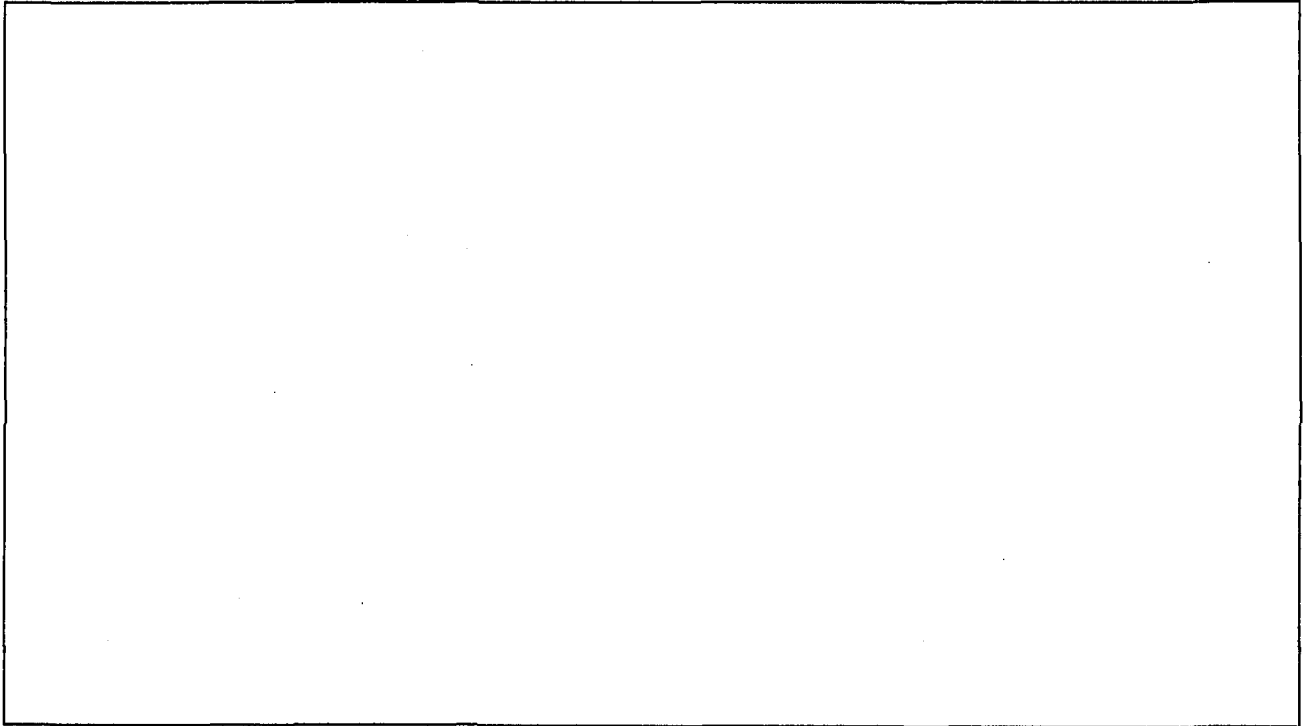


2) return

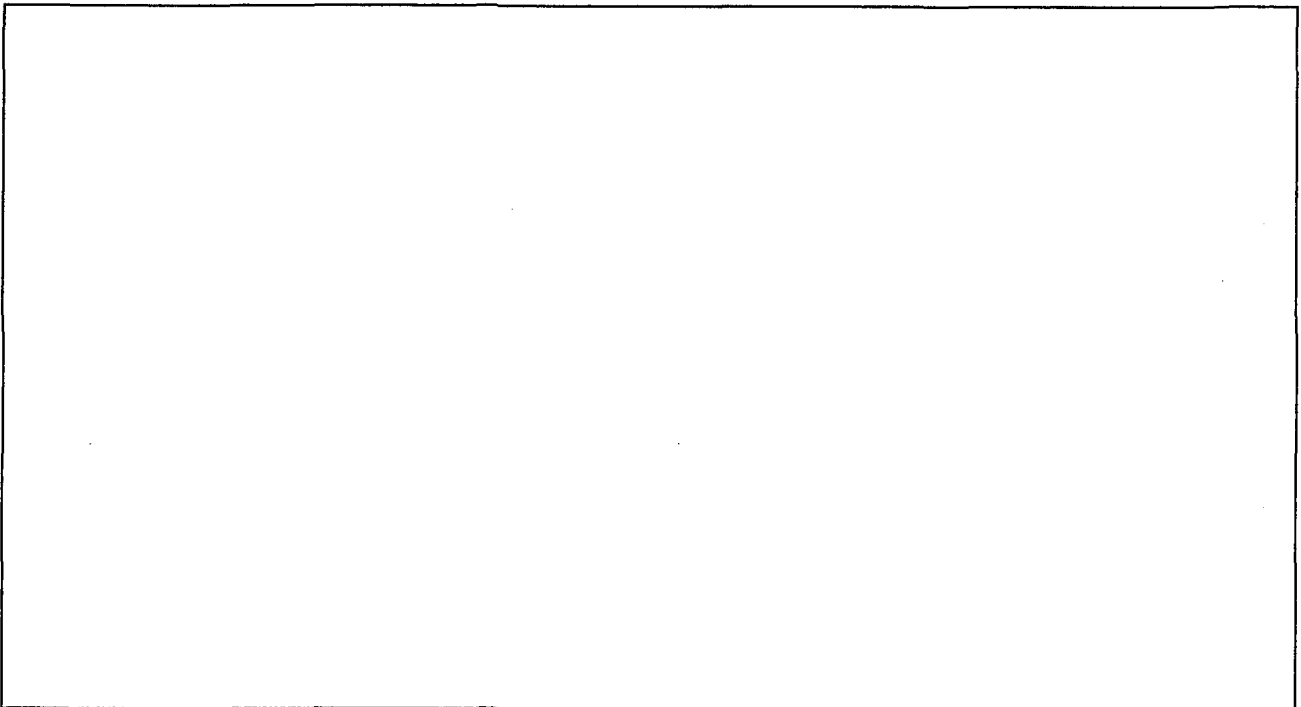


Student ID: .....

3) elif

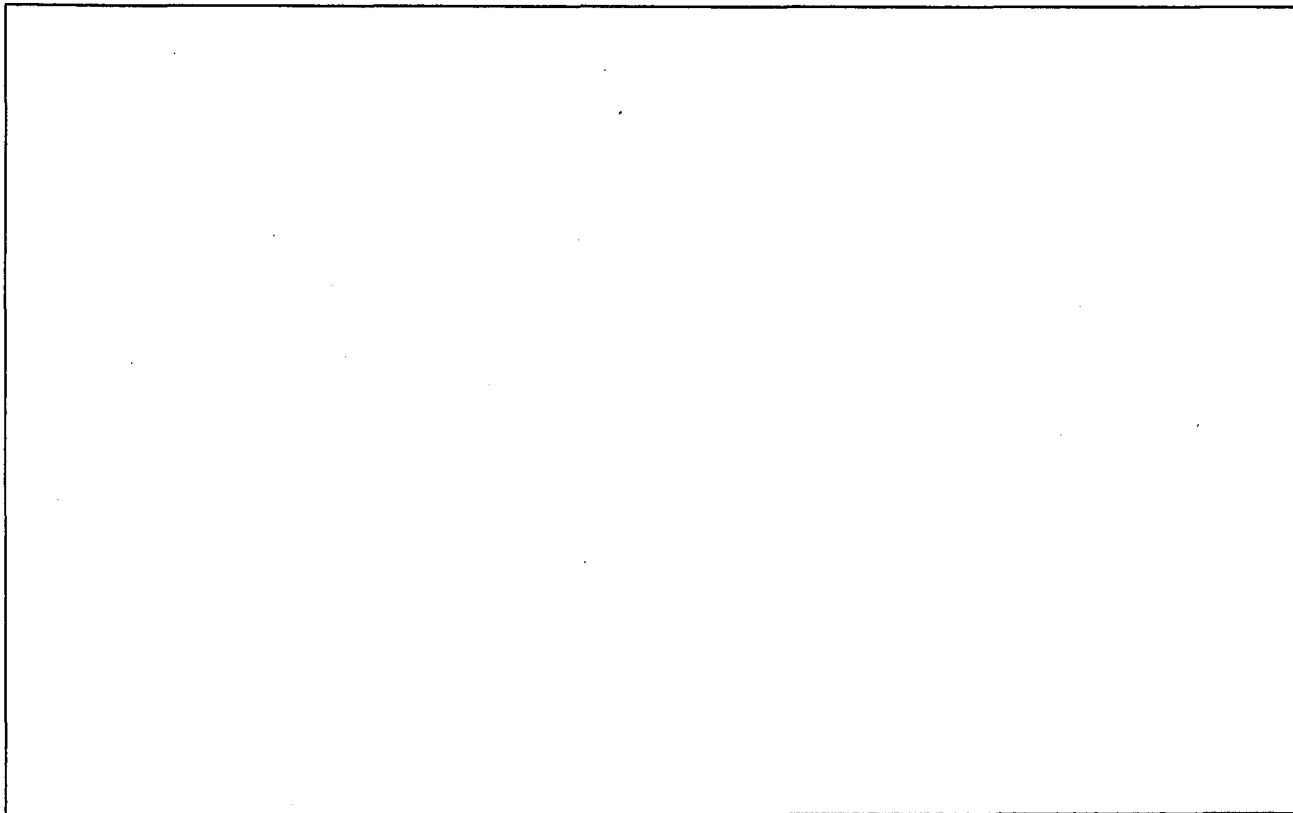


4) in



Student ID: .....

5) and



Student ID: .....

## Question 2 - Reading Python code

[30 marks]

This question concerns the following python program that can be used to convert data from one format to another.

### convertpeers.py:

```
#!/usr/bin/env python

# (1)
import sys
import string
import re

# (2)
(datafile) = sys.argv[1]

as_peer_dict = dict()

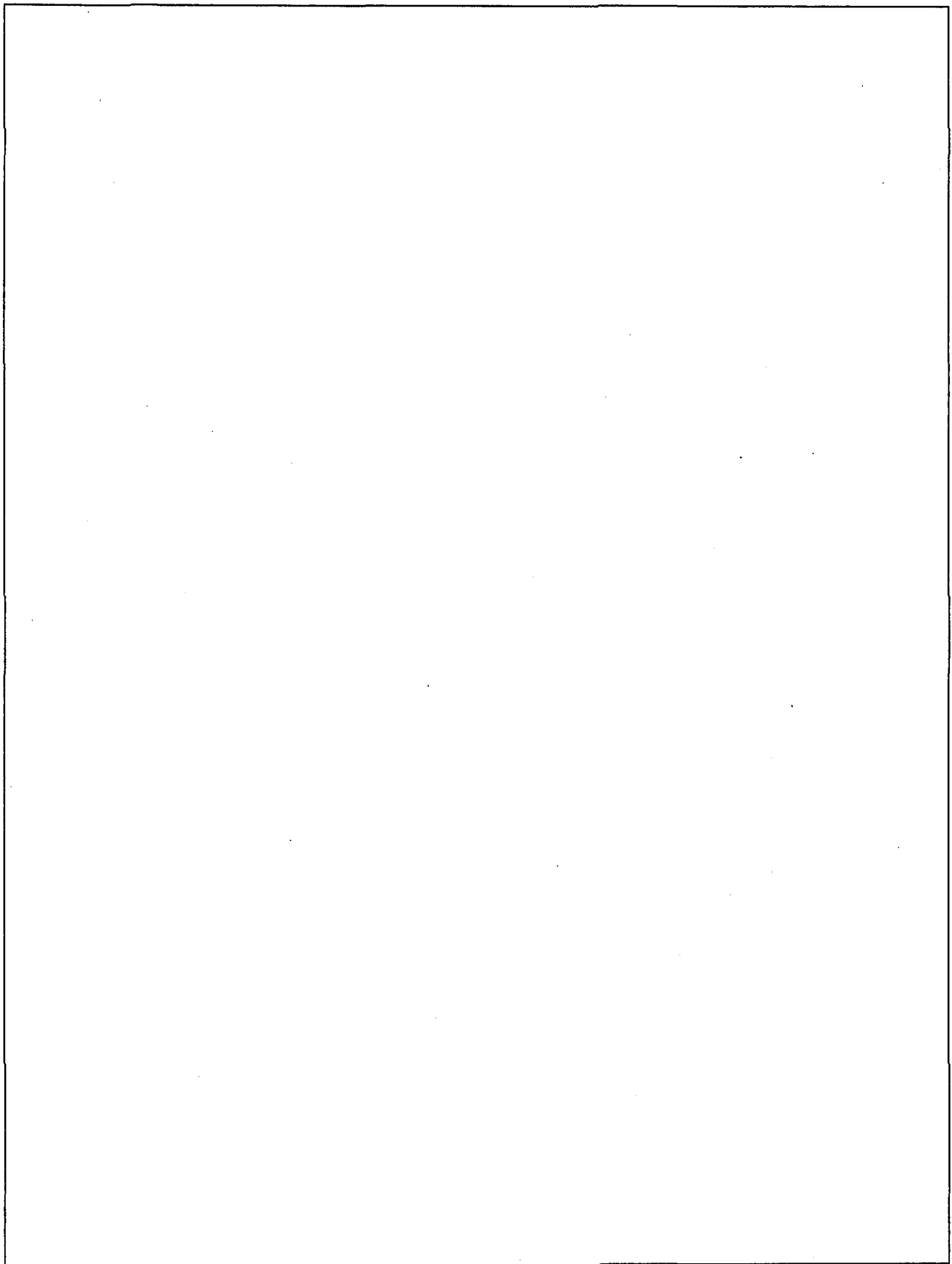
# (3)
fin = open (datafile)

for line in fin:
    # (4)
    (IPaddr, ASnum, desc, md5, status) = line.split(':')
    # (5)
    if re.match('\d*\.\d*', ASnum):
        (hi, lo) = ASnum.split('.')
        ASnum = (int(hi) * 65336) + int(lo)
    # (6)
    status = status.strip()
    if ASnum not in as_peer_dict:
        # (7)
        as_peer_dict[ASnum] = "" <as_peer asnum='%s'>
        <description>%s</description>
        <status>%s</status>
        <md5>%s</md5>
        <ipv4addr>%s</ipv4addr>
    </as_peer>" % (ASnum, desc, status, md5, IPaddr)
    else:
        # (8)
        as_peer_dict[ASnum] = string.replace(as_peer_dict[ASnum],
        ' </as_peer>',
        ' <ipv4addr>%s</ipv4addr>\n </as_peer>' % IPaddr)
# (9)
fin.close()

# (10)
print '<?xml version="1.0"?>\n<as_peers>'
for peer in as_peer_dict:
    print as_peer_dict[peer]
print '</as_peers>'
```

a) [10 marks] Comments have been removed from this code and replaced with numbers. Read the code and create suitable comments explaining the functions of each section of code. Write your comments on the facing page.

Student ID: .....

A large, empty rectangular box with a thin black border, occupying the majority of the page. It is intended for a student to draw or write their answer.

Student ID: .....

b) [20 marks] **convertpeers.py** is used to convert the data in the file **myix.peers** into an XML format. Show the output when the following command is executed:

```
$ python convertpeers.py myix.peers
```

**myix.peers:**

```
202.7.0.1:23754:Citylink:None:passive
202.7.0.6:23754:Citylink:None:passive
202.7.0.17:23754:Cafenet-PS:None:passive
202.7.0.110:131193:VUW-CSW:None:passive
202.7.0.186:2.515:TEC:None:passive
```

**convertpeers.py** is reproduced here for ease of reference:

```
#!/usr/bin/env python

import sys
import string
import re

(datafile) = sys.argv[1]

as_peer_dict = dict()

fin = open (datafile)

for line in fin:
    (IPaddr, ASnum, desc, md5, status) = line.split(':')

    if re.match('\d*\.\d*', ASnum):
        (hi, lo) = ASnum.split('.')
        ASnum = (int(hi) * 65336) + int(lo)
        status = status.strip()

    if ASnum not in as_peer_dict:
        as_peer_dict[ASnum] = "" <as_peer asnum='%s'>
        <description>%s</description>
        <status>%s</status>
        <md5>%s</md5>
        <ipv4addr>%s</ipv4addr>
    </as_peer>"" % (ASnum, desc, status, md5, IPaddr)
    else:
        as_peer_dict[ASnum] = string.replace(as_peer_dict[ASnum],
        ' </as_peer>',
        ' <ipv4addr>%s</ipv4addr>\n </as_peer>' % IPaddr)

fin.close()

print '<?xml version="1.0"?>\n<as_peers>'
for peer in as_peer_dict:
    print as_peer_dict[peer]
print '</as_peers>'
```



Student ID: .....

Write your answer out fully and neatly.

Student ID: .....

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.

Specify the question number for work that you do want marked.

Student ID: .....

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.

Specify the question number for work that you do want marked.

Student ID: .....

**Question 3 - Writing Python programs**

**[30 marks]**

a) [10 marks] Write a Python program that reads two non negative integers as parameters on the command line and prints them out. For example, it might be called by:

```
$ python testprog.py 2 1
```

Your program should check the arguments and issue appropriate error messages and return values.

Student ID: .....

b) [10 marks] What changes would you make to your program to prompt the user for input if no arguments were given on the command line?

Student ID: .....

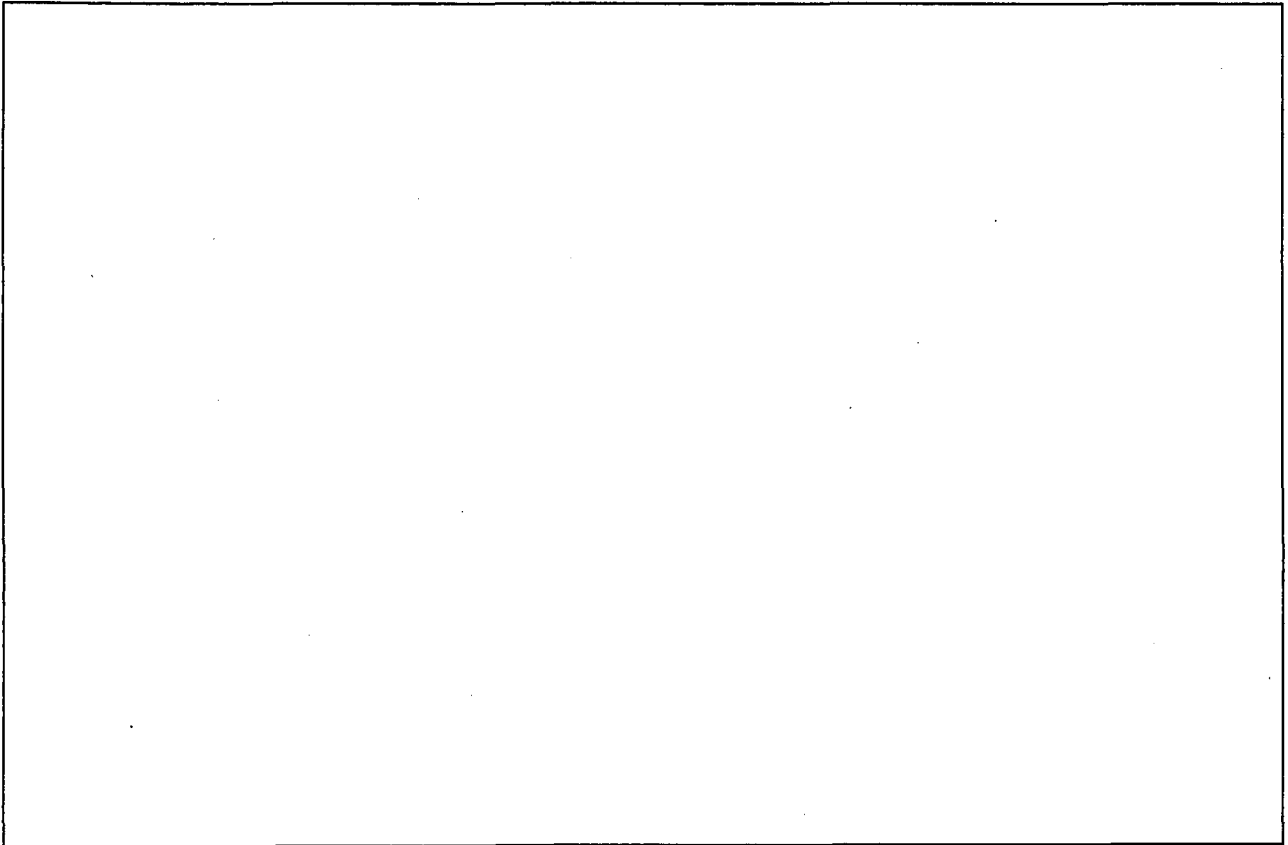
c) [10 marks] The greatest common divisor (GCD) of two integers,  $a$  and  $b$ , is the largest number that divides both of them with no remainder.

You have a systems programming task that requires extensive computation of the GCD of two integers and you need to program this as efficiently as possible.

You have the option of using one of two possible predefined functions  $gcdr(a, b)$ , which uses recursion, and  $gcdw(a, b)$ , which uses a while loop. Both of these return the greatest common divisor.

Describe how you would test these two implementations to see which one would be a better choice. Your answer should consider different factors such as good use of system resources, how the program inputs might affect the results of your tests and the number of iterations of any tests you might try. Other factors may apply and you are encouraged to include these in your answer.

Student ID: .....



Student ID: .....

**Question 4 C Fundamentals**

**[30 marks]**

(a) [4 Marks] Explain the four steps of compilation for C programs.

(b) [4 Marks] State the two most important differences between C and Java according to the lectures.

(c) [4 Marks] Discuss the advantages and disadvantages of iteration versus recursion in C.



Student ID: .....

(d) [4 Marks] Consider the following declaration.

```
int r1, r2, (*fp1)(void), (*fp2)(void), func(void);
```

State whether each of the four statements is correct or incorrect.

```
fp1 = func;
```

```
fp2 = &func;
```

```
r1 = (*fp1)(void);
```

```
r2 = fp2(void);
```

(e) [4 Marks] Consider the following declaration.

```
int a[4][4], (*b)[4], *c[4], **d;
```

Which of a, b, c or d could you pass to a function expecting a pointer to a pointer to int (e.g., `int func(int **);`)? Explain your answer.

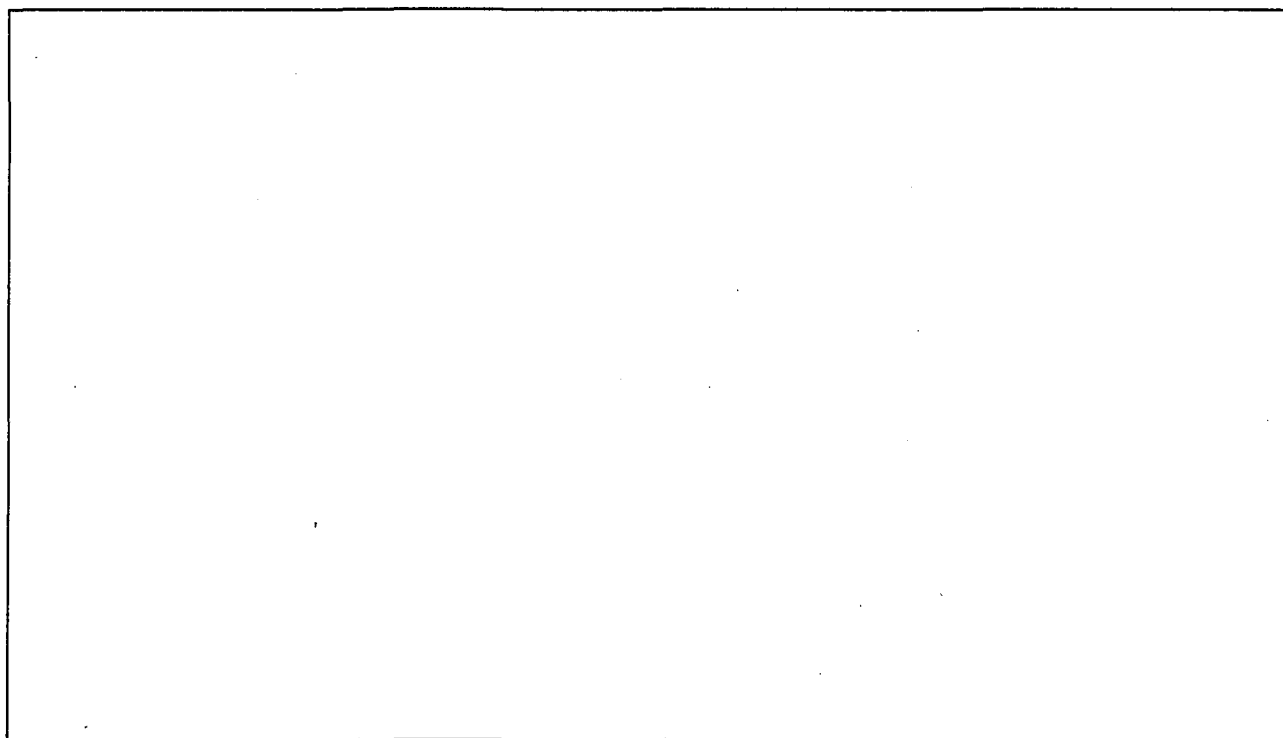
Student ID: .....

(f) [4 Marks] Consider the following struct type:

```
typedef struct {  
    int i;  
    float f;  
} int_float;
```

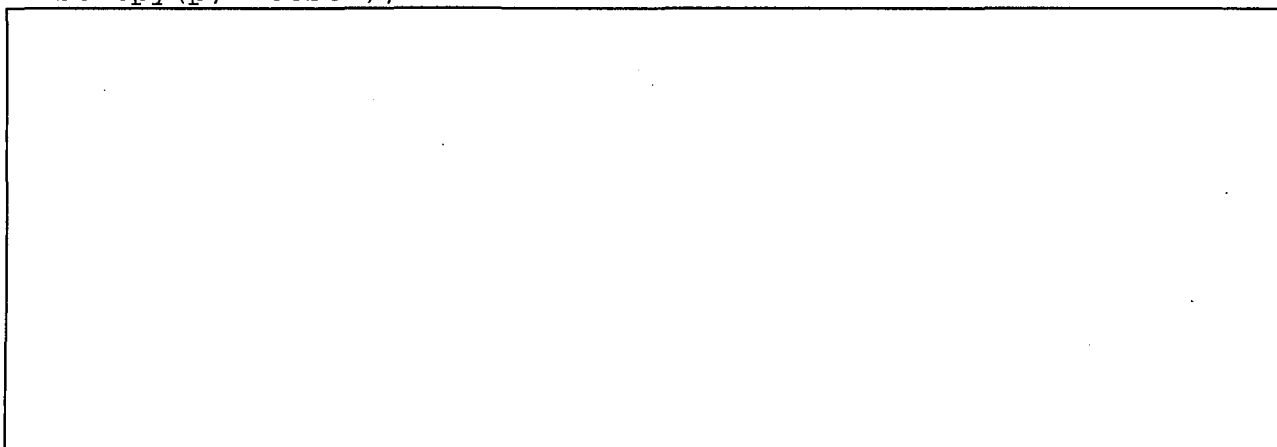
Give a sensible definition of the function `plus()`, which will “add” two `int_float` variables. For example, `plus()` might be used in the following code.

```
int_float a, b, c;  
  
... /* (code to define a and b) */  
  
c = plus(a, b);
```



(g) [3 Marks] State whether the following code is safe and justify your answer.

```
char *p;  
strcpy(p, "test");
```



Student ID: .....

(h) [3 Marks] State whether the following code would work and justify your answer.

```
int *p1 = malloc(128);  
  
int main(void)  
{  
    char *p2 = malloc(128);  
  
    ...  
}
```

Student ID: .....

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.

Specify the question number for work that you do want marked.

Student ID: .....

**Question 5 Arrays and Pointers, File Handling**

**[30 marks]**

- (a) [8 Marks] Using the following function prototype, write a definition of the function `max_diff` that returns the difference between the largest and smallest elements of the first `n` elements of an array-of-float.

```
float max_diff(const float *a, int n);
```

- (b) [6 Marks] Using the following function prototype, write a definition of the function `is_within` that takes a character and a string pointer as its two arguments. The function should return a nonzero value if the character is in the string and zero otherwise.

```
int is_within(const char * str, char ch);
```

Student ID: .....

(c) [6 Marks] Consider the following declaration.

```
char m[4][6] = {"ABCDE", "FGHIJ", "KLMNO", "PQRST"};
```

Give the outputs of the following printf statements.

```
printf("%c", **m);
```

```
printf("%c", *(*m+2));
```

```
printf("%c", *(* (m+1)+1));
```

```
printf("%c", *(m[1]+2));
```

```
printf("%c", (*(m+2))[3]);
```

```
printf("%c", *(&m[0][0]+9));
```

Student ID: .....

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.

Specify the question number for work that you do want marked.

Student ID: .....

- (d) [10 Marks] Write a command-line-arguments based program. The program will be called with two file names as its command line arguments. The program should read the text from the first file and write it to the second file, but removes all occurrences of the letter e (both uppercase and lowercase).

For example, if the first file contains:

```
riverrun, past Eve and Adam's, from swerve of shore to bend of  
bay, brings us by a commodius vicus of recirculation back to  
Howth Castle and Environs.
```

after running your program, the second file should contain:

```
rivrrun, past v and Adam's, from swrv of shor to bnd of bay,  
brings us by a commodious vicus of rcirculation back to Howth  
Castl and nvirons.
```

You need to implement the program with two functions: `main()` and `removeEs()`.

The `main()` function should open and close the files and call the function `removeEs()`.

The `removeEs()` function should read and write the files, removing the letter e as it goes.



Student ID: .....

Student ID: .....

**Question 6 Bitwise Operator and Dynamic Data Structures**

**[30 marks]**

- (a) [8 Marks] Suppose you are working on a 32-bit machine. Write a program that prints each bit of an integer. The program should get the user to type in an integer, and then should print the integer in the following format with spaces between blocks of 8 bits and a new line character at the end.

01001000 01101101 00001111 00010111

- (b) [6 Marks] Write a definition of the function `bitwise_swap` that uses only bitwise operators to swap the values of two integers.

Student ID: .....

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.

Specify the question number for work that you do want marked.

Student ID: .....

- (c) The following type definitions, macro definition and function prototypes are part of a queue model, where we are using singly-linked lists to implement queues.

```
#define Node_Size sizeof(Node)

typedef struct node
{ char data;
  struct node *next;
} Node;

typedef char Data;
typedef Node *ptrNode;

typedef struct queue
{ int cnt; /* counts the number of nodes */
  ptrNode front; /* points to the front node */
  ptrNode rear; /* points to the rear node */
} Queue;

typedef Queue *ptrQueue;

void enqueue(Data, ptrQueue);
Data dequeue(ptrQueue);
```

Suppose the queue that we have implemented has a header node of type Queue and a list of linked nodes of type Node. The pointer front in the header node points to the front node in the list, while the pointer rear points to the rear node in the list. The header node also has a counter cnt, which counts the number of nodes in the list.

- i. [8 Marks] Write C code to implement the function enqueue, which adds a new node to the rear of the list. The character passed to enqueue needs to be assigned to the variable data in the node. You can assume your requests for memory are always successful. You DO NOT need to consider the case when the queue is empty and you may assume the queue is never full.

Student ID: .....

- ii. [8 Marks] Write C code to implement the function `dequeue`, which deletes the front node from the list. The character stored in `data` needs to be returned. You DO NOT need to consider the case when the queue is empty, and you may assume the queue is never full.

\*\*\*\*\*