Exam Question Examples

ENG-270 Computational Methods and Tools (Fall 2025)

Points:

- Each correct MCQ answer for problems 1–10 counts +1 point, incorrect answer counts -0.5 points. No answer counts 0 points. Circle one answer.
- Each correct MCQ answer for problems 11–14 counts +1.5 points, incorrect answer counts -0.75 points. No answer counts 0 points. Circle one answer.
- Each of the coding-on-paper problems 15–16 will be awarded 0 to 5 points (i.e., partial credit is possible). Write your coding-on-paper solutions on this and additional sheets of paper provided.

1 MCQ

Examine the following program in C.

```c
#include <stdio.h>

int main() {
  double b = 1 / 2;
  printf("b = %f\n", b);
  return 0;
}
```

What is most likely printed from the program?

1. b = 0
2. b = 0.000000
3. b = 0.500000
4. None of the above.

2 MCQ

Examine the following program in MATLAB.

```
x = double(450000.5);
y = int32(100000) / 2;

disp(x + y);
```

We would like the result of `x + y` to be 500000.5. Which of the following statements are true?

1. `y` is automatically converted to a floating point number before the addition.
2. The program prints an integer value.
3. None of the above.

3 MCQ

Examine the following program in C.

```c
#include <stdio.h>

int main() {

  double x = 450000.5;
  int y = 100000 / 2;
```

```
  printf("%f\n", x + y);
  return 0;

}
```

We would like the result of `x + y` to be 500000.5. Which of the following statements are true?

1. `y` is automatically converted to a floating point number before the addition.
2. The program prints an integer value.
3. None of the above.

4 MCQ

View the following program in C.

```c
#include <stdio.h>

struct GpsPoint {
  double lon;
  double lat;
  float alt;
};

struct GpsPoint convert_to_km(struct GpsPoint *points, int i) {
  struct GpsPoint point_km;
  point_km.lon = points[i].lon;
  point_km.lat = points[i].lat;
  point_km.alt = points[i].alt / 1e3;
  return &point_km;
}

int main() {
  struct GpsPoint points[2] = {{7.65833, 45.97639, 4478},
                               {6.865, 45.832778, 4808}};
  struct GpsPoint point_km = convert_to_km(points, 1);
  printf("%f\n", point_km.alt);
  return 0;
}
```

Given an array of structure `GpsPoint`, we want to use a function to return a structure with altitude of the `i`-th element of the original array in kilometers. What, if any, are the problems with this code?

1. `return &point_km;` should be `return point_km;`.
2. `struct GpsPoint convert_to_km(struct GpsPoint *points, int i)` should be `struct GpsPoint * convert_to_km(struct GpsPoint *points, int i)`.
3. `struct GpsPoint point_km = convert_to_km(points, 1);` should be `struct GpsPoint * point_km = convert_to_km(points, 1);`.
4. (2) and (3).
5. There is no problem with the code.

5 MCQ

Examine the following program in C.

```c
#include <stdio.h>

struct Sommet {
  char * nom;
  double latitude;
  double longitude;
  float altitude;
};

void assign_Matterhorn(struct Sommet *sommet) {
  sommet.nom = "Matterhorn";
```

```
  sommet.latitude = 45.97639;
  sommet.longitude = 7.65833;
  sommet.altitude = 4478;
}

int main() {
  struct Sommet sommet;
  assign_Matterhorn(&sommet);
  printf("nom = %s\n", sommet.nom);
  return 0;
}
```

We want the program to print `nom = Matterhorn`. What, if any, is the problem with this code?

1. `sommet` is not initialized.
2. `sommet` should be passed by reference and not by value.
3. The `nom` field should not be a pointer.
4. `sommet.nom` should be `sommet->nom` (and the same for the other fields in `sommet`).
5. There is no problem with the code.

6 MCQ

Examine the following program in C.

```
#include <stdio.h>

struct Sommet {
  char * nom;
  double latitude;
  double longitude;
  float altitude;
};

void assign_Matterhorn(struct Sommet sommet) {
  sommet.nom = "Matterhorn";
  sommet.latitude = 45.97639;
  sommet.longitude = 7.65833;
  sommet.altitude = 4478;
}

int main() {
  struct Sommet sommets[1];
  assign_Matterhorn(sommets[0]);
  printf("nom = %s\n", sommets[0].nom);
  return 0;
}
```

We want the program to print `nom = Matterhorn`. What, if any, is the problem with this code?

1. `sommets` is not initialized
2. `sommets[0]` should be passed by reference and not by value
3. The `nom` field should not be a pointer.
4. The `sommet` variable in the function `assign_Matterhorn` should be called `sommets`.
5. There is no problem with this code.

7 MCQ

Examine the following program in C.

```
#include <stdio.h>

int square(int x) {
  x = x * x;
  int y = x;
  return y;
};
```

```c
int main() {
  int x[] = {2};
  int y = square(x[0]);
  int z = y/x[0];
  printf("z = %d\n", z);
  return 0;
}
```

What is the printed output?

1. `z = 1`
2. `z = 2`
3. `z = 4`
4. `z = 1.00000`
5. `z = 2.00000`

8 MCQ

Examine the following program in C.

```c
#include <stdio.h>

int square(int x) {
  x = x * x;
  int y = x;
  return y;
};

int main() {
  int x = 2;
  int y = square(x);
  int z = y / x;
  printf("z = %d\n", z);
  return 0;
}
```

What is the printed output?

1. `z = 1`
2. `z = 2`
3. `z = 4`
4. `z = 2.00000`

9 MCQ

Examine the following program in C.

```c
#include <stdio.h>

int square(int * x) {
  *x = *x * *x;
  int y = *x;
  return y;
};

int main() {
  int x = 2;
  int y = square(&x);
  int z = y / x;
  printf("z = %d\n", z);
  return 0;
}
```

What is the printed output?

1. `z = 1`

2. `z = 2`
3. `z = 4`
4. `z = 2.00000`

10 MCQ

Examine the following program in MATLAB.

```
x = [2];
y = square(x);
z = y / x;

fprintf('z = %d\n', int16(z));

function y = square(x)
  x(1) = x(1).^2;
  y = x(1);
end
```

What is the printed output?

1. `z = 1`
2. `z = 2`
3. `z = 1.00000`
4. `z = 2.00000`
5. `z = 32767`

11 MCQ

Examine the following program in C.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#define nCapteur 2

struct Capteur {
  double x;
  double y;
};

double * dist(double x, double y, struct Capteur * capteurs) {
  double * distance = malloc(sizeof(double) * nCapteur);
  for (int s = 0; s < nCapteur; s++) {
    double dx = x - capteurs[s].x;
    double dy = y - capteurs[s].y;
    distance[s] = sqrt(dx * dx + dy * dy);
  }
  return distance;
}

int main() {
  struct Capteur capteurs[nCapteur];

  capteurs[0].x = 7340;
  capteurs[0].y = 4020;

  capteurs[1].x = 7500;
  capteurs[1].y = 2880;

  double x = 7000.;
  double y = 4000.;
  double * distance = dist(x, y, capteurs);

  for(int s = 0; s < nCapteur; s++) {
    printf("s = %d, distance = %f\n", s, distance[s]);
```

```
  }
  free(distance);

  return 0;
}
```

We would like to print the distances from the designated point to each of the sensors. What, if any, are the problems with this code?

1. You cannot return an array from a function.
2. In the argument list, `struct Capteur * capteurs` should be replaced by `struct Capteur capteurs`.
3. `return distance;` must be replaced by `return &distance;`
4. (2) and (3).
5. There is no problem with this code.

12 MCQ

Recall the cryptogramme exercise, where the message encoded in `cryptogramme` must be deciphered. The instructions were as follows:

"The first letter goes in box 17. For each symbol that follows, advance 17 positions, but only count the empty positions. If you get to the end of the string of text, start over at the beginning."

```c
#include <stdio.h>
#define NCHAR 56
#define NADV 17

int main() {

  char *cryptogramme = "T-LS-AOS--EEOIOAAUENTUOLUILRPDSA-S.LUNV-ENEGT-T-E-NLLSBE";

  char *message = "*******************************************************\0";

  int pos = 0;
  for (int i = 0; i < NCHAR; i++) {
    int adv = 0;
    while(adv < NADV) {
      pos = (pos + 1) % NCHAR;
      if(message[pos] == '*') adv++;
    }
    message[pos] = cryptogramme[i];
  }
  message[NCHAR] = '\0';

  printf("Message: %s\n", message);

  return 0;
}
```

What, if any, are the problems with this code?

1. The length of `message` string array should be 56 and not 57.
2. `message` should be declared as a string array (`char message[]`) and not a pointer to a string (`char *message`).
3. (1) and (2).
4. There is no problem with this code.

13 MCQ

Below is an alternate solution to the cryptogramme problem (reintroduced in the previous problem).

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

```c
#define NADV 17

int main() {

  char *cryptogramme = "T-LS-AOS--EEOIOAAUENTUOLUILRPDSA-S.LUNV-ENEGT-T-E-NLLSBE";

  int nchar = strlen(cryptogramme);
  char *message = malloc(sizeof(char) * (nchar + 1));
  // Initialiser le message à ******...
  for (int i  = 0; i < nchar; i++) {
    message[i] = '*';
  }
  message[nchar + 1] = '\0';

  int pos = 0;
  for (int i = 0; i < nchar; i++) {
    int adv = 0;
    while(adv < NADV) {
      pos = (pos + 1) % nchar;
      if(message[pos] == '*') adv++;
    }
    message[pos] = cryptogramme[i];
  }
  message[nchar] = '\0';
  free(message);

  printf("Message: %s\n", message);

  return 0;
}
```

What, if any, are the problems with this code?

1. The length of `message` string array should be `size + 1`.
2. `message` should be declared as a string array (`char message[NCHAR+1]`) and not a pointer to a string (`char *message`).
3. Memory for `message` is freed in the wrong location.
4. (1), (2), and (3).
5. There is no problem with this code.

14 MCQ

Recall the "seismes" exercise. Given the seismic signal, an earthquake is said to occur if the absolute value of the signal exceeds a value of 20 and the duration is determined when 200 consecutive values are below the threshold.

The program below finds the beginning and end of each earthquake event. The contents of `lireFichier` are omitted (`{...}`) for brevity but reads the file `nomFichier` and writes data to the array `tableauAremplir` up to the `longueur` provided to the function.

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int lireFichier(char * nomFichier, double * tableauARemplir, int longueur) {
  {...}
  return n;
}

void debut_et_fin(int longueur, double *tremblement, int *i, int *debut, int *fin) {
  while (*i < longueur) {
    if (fabs(tremblement[*i]) > 20) break;
    *i += 1;
  }
  *debut = *i;

  // Fin
  int compteur = 0;
```

```
  while (*i < longueur) {
    if (fabs(tremblement[*i]) < 20) {
      // On ajoute 1, et vérifie si on a atteint 200
      compteur += 1;
      if (compteur == 200) break;
    } else {
      // On remet le compteur à 0
      compteur = 0;
    }
    *i += 1;
  }
  *fin = *i - 199;
  return;
}

int main(int argc, char * argv[]) {
  double tremblement[11000];
  int longueur = lireFichier("earthquake-multiple", tremblement, 11000);

  int i = 0;
  int debut = 0;
  int fin = 0;
  while (i < longueur) {

    debut_et_fin(longueur, tremblement, &i, &debut, &fin);
    if(debut == longueur) break;

    printf("Début à %d, fin à %d, longueur %d\n", debut, fin, fin - debut);
  }

  return 0;
}
```

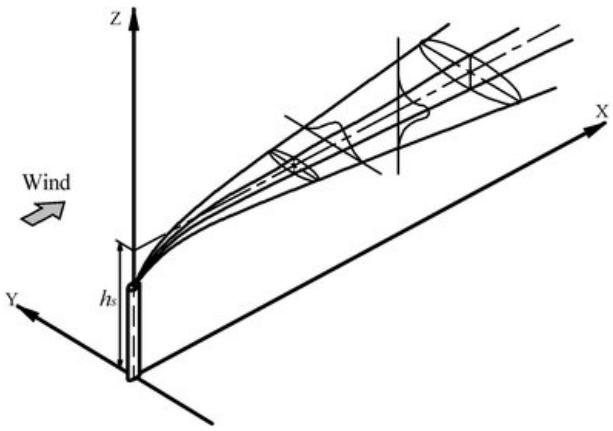
The function `debut_et_fin` finds the beginning (`debut`) and (`fin`) end of each earthquake event. What, if any, are problems with this code?

1. The address of index `i` and not its value is passed to `debut_et_fin`.
2. In `debut_et_fin`, `*i` should be replaced by `i`.
3. `debut` and `fin` are not returned from the function.
4. (1) and (2).
5. There is no problem with this code.

15 Coding-on-paper Problem

Write a program to calculate a person's integrated exposure to the pollution as they ride a bicycle through an air polluiton plume.

A Gaussian plume model for dispersion of air is illustrated below (from Liu et al. `https://doi.org/10.1007/s11356-015-5404-8`, 2015).



The plume model assumes that under constant emission rate Q and meteorological conditions, the mean concentration field remains unchanged. The concentration C is described as a function of position (x, y, z):

$$C(x,y,z) = \frac{Q}{2\pi u\,\sigma_y\,\sigma_z} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \exp\left(-\frac{(z-h_s)^2}{2\sigma_z^2}\right)$$
$$\sigma_y = ax(1+0.0001x)^{-1/2}$$
$$\sigma_z = px(1+qx)^r \ .$$

The emission source is located at the coordinate $x = 0, y = 0, z = h_s$, where h_s is the height of the stack. x is taken to be positive in the direction of the the wind; u is the magnitude of wind speed in this direction. σ_y and σ_z (in meters) are width parameters that describe how the Gaussian profile spreads in the lateral and vertical direction perpendicular to that of propagation and increase with downwind distance x (also in meters). The parameters a, p, q, r are dependent on atmospheric stability conditions.

The person is traveling through the plume on bicycle from point (x_1, y_1) to (x_2, y_2) at a constant velocity v. You can assume that $z = 0$. This is a case of linear interpolation that can be parameterized by time t:

$$x(t) = x_1 + (x_2 - x_1)\frac{v \cdot t}{d}$$
$$y(t) = y_1 + (y_2 - y_1)\frac{v \cdot t}{d}$$
$$\text{where} \quad d = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2} \ .$$

The integrated exposure is calculated as

$$E = \int_{T_1}^{T_2} C(x(t), y(t))dt \ .$$

You can approximate this interval by assuming the concentration to be constant across a series of discrete intervals $\{t_1 = T_1, t_2, \ldots, t_k, \ldots, t_N = T_2\}$ separated by Δt.

$$E \approx \sum_{k=1}^{N} C(x(t_k), y(t_k))\Delta t \ .$$

Write a program in MATLAB or pseudo-code calculates the integrated exposure over the time the student rides from point (x_1, y_1) to (x_2, y_2). Show how the program can be used to calculate E for values $(x_1 = 5000, y_1 = -2000)$, $(x_2 = 7000, y_2 = 3000)$, $Q = 60$ kilograms per second, $u = 5$ meters per second, $h_s = 60$ meters, and dispersion parameters $a = 0.11, p = 0.08, q = 0.0002, r = -0.5$. (You do not have to actually compute the numerical answer.)

16 Coding-on-paper problem

Calculate the predominant wind direction for each month.

Below are few lines of a file (called "LAUSANNE_wind.csv") containing wind directions and wind speeds in 10-minute intervals in Lausanne (from MeteoSwiss and the Swiss National Air Pollution Monitoring Network).

```
2022 01 01 00 10     73.50      0.41
2022 01 01 00 20     90.85      0.35
2022 01 01 00 30     89.17      0.69
2022 01 01 00 40    138.99      0.63
2022 01 01 00 50    121.33      0.79
2022 01 01 01 00     96.49      1.03
...
2022 12 31 23 00    279.87      2.09
2022 12 31 23 10    291.25      2.14
2022 12 31 23 20    306.84      2.07
2022 12 31 23 30    313.46      1.64
```

```
2022 12 31 23 40    301.56      1.69
2022 12 31 23 50    312.10      1.32
2022 12 31 24 00    316.01      1.41
```

The full file contains 52560 observations of 7 variables in columns separated by a whitespace:

1. Year
2. Month
3. Day
4. Hour
5. Minute
6. Wind direction in degrees
7. Wind speed in meters per second (m/s)

To interpret the wind directions:

- The wind was blowing from the north (N) when the direction was between 315 and 45 degrees.
- The wind was blowing from the east (E) when the direction was between 45 and 135 degrees.
- The wind was blowing from the south (S) when the direction was between 135 and 225 degrees.
- The wind was blowing from the west (W) when the direction was between 225 and 315 degrees.

Using MATLAB syntax and vectorization where possible, write a program to calculate the predominant (i.e., most frequent) wind direction for each month, discarding wind speeds below 2 m/s (wind directions are unreliable below this speed). Print the following statements where the results of your calculation should fill in the blanks "_" for the direction (N, S, E, W).

The predominant wind direction in month 1 was _.

The predominant wind direction in month 2 was _.

. . . and so on for all 12 months.

You can assume that you are provided a function that returns each of the fields above as row matrices of dimensions 1×52560:

```
[year, month, day, hour, minute, direction, speed] = readfile('LAUSANNE_wind.csv');
```

where the first 5 variables are of type `int16` and the last two are of type `double`.