

**Mid-Term Test**  
**Practical Portion**

EEE243 Applied Computer Programming  
22 October 2015, 0800–1050 hrs

**Examiner:** W. Greg Phillips, CD, PhD, PEng

**Test de mi-session**  
**Partie Pratique**

GEF243 Programmation informatique appliquée  
22 octobre 2015, 8h00–10h50

**Examineur:** Capt Adrien Lapointe, CD, MSc

**Instructions**

- You may begin the practical portion of the exam immediately.
- The practical portion is out of 30 marks.
- The practical portion of this test is open book. You may refer to course notes and you may access the Internet.
- Communication with any person other than the invigilators during the practical portion of the test **is forbidden**.
- You will not receive an exam booklet for the practical portion of the test; you have received three pages of notepaper instead.
- Write your name and student number on the notepaper, and turn it in at the end of the exam if you wish it to be marked by the examiner.
- Ask the invigilator for more notepaper if you require it.

**Hint:** Compile and execute your program **often**. You will lose marks if your program does not compile and execute as submitted, or if it generates compiler warnings. Start small and build from there.

**Instructions**

- Vous pouvez commencer la partie pratique tout de suite.
- La valeur de la partie pratique est de 30 points.
- La partie pratique de l'examen est à livre ouvert. Vous pouvez faire référence aux notes du cours et accéder à l'Internet.
- La communication avec toutes personnes autre que les surveillants pendant la partie pratique de l'examen **est interdite**.
- Vous ne recevrez pas de livret d'examen pour la partie pratique du test; vous avez plutôt reçu trois feuilles de notes.
- Écrivez votre nom et numéro de collège sur le papier de notes et remettez-le à la fin de l'examen si vous voulez qu'elles soient corrigées par l'examineur.
- Demandez plus de feuilles de notes si vous en avez besoins.

**Indice :** Compilez et exécutez votre code **souvent**. Vous serez pénalisé si votre programme ne compile pas, n'exécute pas ou génère des avertissements de compilateur. Commencez petit et ajoutez des fonctionnalités au fur et à mesure.

## Introduction

Open the KobayashiMaru project that you created at the beginning of the examination.

**Add your name and college number** to the comment blocks at the beginnings of the KobayashiMaru.c and hulldamage.c files.

The code provided is the skeleton of a simulation for hull damage to the spaceship Kobayashi Maru. A section of hull is represented by the square array hull, where each element represents an area of the ship's hull. When a missile strikes the hull, it causes damage to the point of impact and lesser damage to the area around it.

Your job is to write a part of the simulation program. This includes a function to represent the current state of the hull as grey scale "ASCII art", a function to compute hull strength following a missile impact, and a function to determine whether the hull has been breached.

**You must provide evidence of your design** for each part of the tasks. Clear, well-structured code that works and includes appropriate comments is sufficient. However, if your code does not work or is not clear or well structured, you may receive part marks for design documentation in the form of flowcharts, pseudo-code, memory diagrams, or textual descriptions of your design. Write these on the provided notepaper.

## Introduction

Ouvrez le projet KobayashiMaru que vous avez créé au début de l'examen.

**Ajoutez votre nom et numéro de collège** au bloc commentaire au début des fichiers KobayashiMaru.c. et hulldamage.c.

Le code fourni est le squelette d'une simulation des dommages à la coque du vaisseau spatial Kobayashi Maru. Une section de coque est représentée par le tableau carré hull où chaque élément est une portion de la coque du vaisseau. Lorsqu'un missile frappe la coque, il endommage le point d'impact et cause des dommages moindres autour.

Votre tâche est d'écrire une partie du programme de simulation. Cela inclut une fonction pour représenter l'état courant de la coque en tant qu'échelle de gris en « art ASCII », une fonction pour calculer la force de la coque et une fonction pour déterminer si la coque a été percée.

**Vous devez fournir des traces de votre conception** pour chaque partie des tâches. Du code clair et bien structuré qui fonctionne et qui inclut des commentaires appropriés est suffisant. Toutefois, si votre code ne fonctionne pas ou s'il n'est pas clair ou bien structuré, vous pouvez recevoir des points partiels pour la documentation de conception sous la forme d'organigrammes, pseudo-code, schémas de mémoire ou descriptions textuelles de votre conception. Écrivez-les sur les feuilles de notes fournies.

## Requirements

You have been provided with three files: KobayashiMaru.c, hull\_damage.c and hull\_damage.h.

In hull\_damage.c you must implement the three functions `convert`, `missile_impact`, and `breached`.

In KobayashiMaru.c you must write code that simulates a series of missile impacts, displaying the state of the hull after each impact as an “ASCII-art” grey scale image. You may also implement any other functions that you judge necessary.

You **must not modify** hull\_damage.h.

## Hull representation

The ship’s hull is represented by a square, two-dimensional array of integers, each of which represents the strength of an area of the ship’s hull. Initially, all elements of the hull have `INITIAL_HULL_STRENGTH` (see hull\_damage.h). The hull’s minimum strength is zero, which represents a hole in the hull. The size of the hull is specified by `EDGE_LENGTH` in hull\_damage.h.

### main

While working through this exam, you should use the `main` function in KobayashiMaru.c to test the functionality of the `convert`, `missile_damage` and `breached` functions described below.

When you are done, the `main` function must:

- Initialize the hull using the `initialize` function.
- Display the hull’s strength as ASCII-art, using the `convert` and `display_characters` functions.
- The array `impacts` consists of a list of pairs, each representing the row and column of a missile impact. For each impact in the array:
  - simulate its effects using the `missile_impact` function, then
  - display the resulting hull strength as ASCII art, and
  - check whether the hull has been breached using the `breached` function, and if so, print an appropriate message.

### convert

The `convert` function takes two parameters: the `hull` array and a two dimensional array of characters called `greys`, which has the same dimensions as `hull`. For each element in `hull`, the `convert` function takes its numeric value, converts it to a character as described below, and puts that character in the corresponding element of `greys`.

## Exigences

Trois fichiers vous sont fourni : KobayashiMaru.c, hull\_damage.c et hull\_damage.h.

Dans hull\_damage.c, vous devez implémenter les trois fonctions `convert`, `missile_impact` et `breached`.

Dans KobayashiMaru.c, vous devez écrire du code qui simule une série d’impact de missiles et qui montre l’état de la coque après chaque impact comme une image de niveau de gris en « art ASCII ». Vous pouvez aussi implémenter toute fonction que vous jugez nécessaire.

Vous **ne devez pas modifier** hull\_damage.h.

## Représentation de la coque

La coque du vaisseau est représentée par un tableau d’entiers carré à deux dimensions où chaque entier représente la force d’une portion de la coque. Au départ, les éléments de la coque sont à `INITIAL_HULL_STRENGTH` (voir hull\_damage.h). La force minimale de la coque est zéro, qui représente un trou. La taille de la coque est spécifiée par `EDGE_LENGTH` dans hull\_damage.h.

### main

Pendant que vous travaillez sur le code, vous devriez utiliser la fonction `main` de KobayashiMaru.c afin de tester les fonctionnalités des fonctions `convert`, `missile_damage` et `breached` décrites ci-dessous.

Lorsque vous aurez terminé, la fonction `main` doit :

- Initialiser la coque en utilisant la fonction `initialize`.
- Afficher la force de la coque comme art ASCII en utilisant les fonctions `convert` et `display_characters`.
- Le tableau `impacts` consiste en une liste de paires représentant chacune la rangée et colonne de l’impact du missile. Pour chaque impact dans le tableau :
  - simulez son effet en utilisant la fonction `missile_impact`, puis
  - affichez la force de la coque résultante comme art ASCII, et
  - vérifiez si la coque a été trouée en utilisant la fonction `breached`. Si elle est trouée, affichez un message approprié.

### convert

La fonction `convert` prend deux paramètres: le tableau `hull` et un tableau de caractères à deux dimensions appelé `greys` qui a les mêmes dimensions que `hull`. Pour chaque élément de `hull`, la fonction `convert` prend sa valeur numérique, la convertit en caractère tel que décrit ci-dessous et met le caractère dans l’élément correspondant de `greys`.

The conversion uses the characters specified in GREYSCALE in hull\_damage.h and maps numeric values to characters as shown below.

```

0 : ' '
1..10 : '.'
11..20 : ':'
21..30 : '-'
... and so on by tens up to
81..90 : '@'

```

An example conversion result is shown below. On the left are the numeric strengths of the hull after one missile has hit at location (5, 5), and on the right is the conversion.

```

90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90
90 90 90 90 80 68 80 90 90 90
90 90 90 90 68 40 68 90 90 90
90 90 90 90 80 68 80 90 90 90
90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90

```

### missile\_impact

The `missile_impact` function simulates the effect of a missile striking the hull. Missile strikes decrease the hull strength at the point of impact and also in the immediately adjacent region. The amounts of strength decrease are given by the constant `DAMAGE` in `hull_damage.h`. The hull's strength is reduced by 50 units at the point of impact; one row above, below, left and right the strength is reduced by 22 units; and on the immediate diagonals the strength is reduced by 10 units. In the example above, the hull was initially at full strength (90 everywhere) when a missile struck at location (5, 5). The hull strength cannot go below zero units at any point.

Take care not to go outside the edges of the hull! The edges of the hull do not “wrap”, so the top edge does not connect to the bottom and the left does not connect to the right.

### breached

The `breached` function takes the `hull` as a parameter and returns true if the strength of any element of the hull has been reduced to zero, false otherwise.

## When you are finished

Ensure your name is in the header comment of the `KobayashiMaru.c` and `hull_damage.c` files and on any notes pages. Leave the computer logged in and your

La conversion utilise les caractères spécifiés dans GREYSCALE de `hull_damage.h` et fait la correspondance entre les valeurs numériques et les caractères, tel qu'illustré ci-dessous

```

0 : ' '
1..10 : '.'
11..20 : ':'
21..30 : '-'
... et ainsi de suite en incrément de dix
81..90 : '@'

```

Un exemple de résultats de conversion est montré ci-dessous. Les valeurs numériques de la force de la coque après un impact à l'emplacement (5, 5) sont à gauche, la conversion est à droite.

```

@ @ @ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @ @ @
@ @ @ @ % # % @ @ @
@ @ @ @ # = # @ @ @
@ @ @ @ % # % @ @ @
@ @ @ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @ @ @

```

### missile\_impact

La fonction `missile_impact` simule l'effet d'un missile frappant la coque. L'impact de missiles diminue la force de la coque au point d'impact et aussi dans les régions immédiatement adjacentes. La magnitude de la diminution de force est donnée par la constante `DAMAGE` dans `hull_damage.h`. La force de la coque est réduite de 50 unités au point d'impact, 22 unités aux rangées immédiatement adjacentes en haut, en bas, à gauche et à droite, et de 10 unités sur les diagonales immédiates. Dans l'exemple ci-dessus, la coque était à sa force maximale (90 partout) lorsqu'un missile a frappé à la position (5, 5). La force de la coque ne peut descendre sous zéro en aucun cas.

Assurez-vous de ne pas sortir des limites de la coque ! Les bords de la coque ne « boucle » pas, c'est-à-dire que le bord du haut ne connecte pas avec celui du bas, celui de gauche ne connecte pas avec celui de droite.

### breached

La fonction `breached` prend le `hull` en paramètre et retourne vrai si la force de n'importe quel élément de la coque (`hull`) a été réduite à zéro. Sinon, elle retourne faux.

## Lorsque vous avez terminé

Assurez-vous que votre nom est dans le bloc commentaire des fichiers `KobayashiMaru.c` et `hull_damage.c`, ainsi que sur vos pages de notes.

project open in Eclipse, and leave your notes pages on the keyboard. Leave quietly by the closest door to minimize disturbance to others.

**End of the examination**

Laissez l'ordinateur connecté, votre projet ouvert dans Eclipse et laissez vos pages de notes sur le clavier. Sortez discrètement par la porte la plus proche pour ne pas déranger les autres.

**Fin de l'examen**