CORRECTION

☐0 ☐0 ☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre, et inscrivez votre nom et prénom ci-dessous.

Nom : .....

Prénom : .....

LIFAP4 Conception et développement d'applications : Examen (CC mi-parcours) du 6 mars 2018 (90 minutes)

Aucun document autorisé, sans calculatrice, sans téléphone

Attention : noircir complètement à l'encre la ou les cases cochées (une croix ne suffit pas). Les questions marquées d'un trèfle peuvent avoir 0, 1 ou plusieurs choix corrects. Ce sera le cas de toutes les questions dans ce sujet.

Question 1 ♣ La commande 'g++ –Wall –ggdb –c main.cpp –I/usr/include'

- ■ '-c' demande la production de fichiers objets (.o).
- ■ '–ggdb' indique d'inclure les informations sur le programme pour le débogueur.
- ☐ '-I/usr/include' indique un chemin vers les bibliothèques (.a, .so ou .dll).
- ☐ '–ggdb' indique d'exécuter le programme avec gdb.
- ☐ effectue la phase d'édition de lien.
- ■ effectue la phase de compilation.
- ■ '-Wall' provoque l'affichage de tous les messages de prévention (ambigüités, type cast, oublis, etc.).
- ■ '-I/usr/include' indique un chemin vers les fichiers d'entête (.h).
- ☐ crée un exécutable.

Question 2 ♣ Avec un débogueur je peux

- ☐ corriger les erreurs de compilation.
- ☐ lancer une série de tests dans un script.
- ☐ obtenir des statistiques d'utilisation de chaque fonction.
- ■ comprendre ce que fait réellement le programme.
- ■ mettre des points d'arrêts.
- ■ inspecter des variables au cours de l'exécution.
- ☐ valider des correctifs.
- ☐ retrouver l'historique des exécutions.

Question 3 ♣ SDL permet facilement

- ■ de récupérer les touches claviers enfoncées par l'utilisateur.
- ☐ de faire saisir à l'utilisateur du texte.
- ■ de jouer des sons.
- ■ d'afficher des images.
- ☐ d'ajouter des menus à une application.

Question 4 ♣ Doxygen

- [x] utilise la grammaire du langage dans lequel est écrit le code source, ainsi que les commentaires.
- [x] est un générateur de documentation sous licence libre capable de produire une documentation à partir du code source d'un programme.
- [ ] récupère et traite des actions de la souris.
- [ ] gère les versions du code d'un programme.

Question 5 ♣ Ce code

```
int *a;
a = 12;
if (*a==12) cout«"Vrai";
```

- [ ] provoquera une erreur de compilation.
- [x] peut provoquer un 'segmentation fault (core dump)'.
- [x] provoquera une erreur avec Valgrind.
- [x] peut provoquer une erreur avec un débogueur.
- [x] peut provoquer une erreur à l'exécution.

Question 6 ♣ Cochez les affirmations vraies à propos d'un diagramme de Gantt.

- [x] La ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois etc.).
- [x] La colonne de gauche du diagramme énumère toutes les tâches à effectuer.
- [x] Un diagramme de Gantt permet de visualiser dans le temps les diverses tâches composant un projet.
- [ ] Un diagramme de Gantt indique les pénalités en cas de retard dans le développement.

Question 7 ♣ Avec Git, la copie de travail

- [ ] contient uniquement des fichiers gérés par Git.
- [ ] se trouve sur le serveur Git.
- [x] est l'ensemble des fichiers sur mon disque dur sur lesquels je travaille directement.
- [ ] se trouve dans le répertoire '.git'.

Question 8 ♣ Avec Git quelle est l'opération inverse de 'git add' ?

- [ ] rm
- [x] git rm fichier
- [ ] git reset HEAD

Question 9 ♣ Je veux supprimer un fichier de mon historique de Git, :

- [ ] j'utilise la commande 'rm' de mon terminal.
- [ ] j'utilise la commande 'git rm'.
- [x] je ne peux pas.

Question 10 ♣ Avec Git, le dépôt distant

- [x] se trouve sur le serveur Git.
- [ ] est l'ensemble des fichiers sur mon disque sur lequel je travaille.
- [ ] se trouve dans le répertoire '.git'.

Question 11 ♣ Pour comprendre, inspecter et déboguer la gestion de la mémoire d'un programme je peux utiliser

- [ ] Git.
- [x] Valgrind.
- [x] un débogueur.
- [ ] les erreurs de g++.
- [x] des 'cout/printf' dans les constructeurs et destructeurs.
- [ ] Doxygen.

Question 12 ♣ La fonction 'SDL_RenderCopy' de SDL2
- ☐ copie une surface sur la fenêtre principale ou tout autre surface cible.
- ■ permet de sélectionner une partie de la texture source et une taille du rectangle destination.
- ■ copie une texture sur la fenêtre principale ou tout autre texture cible.

Question 13 ♣ Avec Git, la commande 'git push'
- ☐ permet d'envoyer les changements des fichiers de travail dans le dépôt local et dans le dépôt distant.
- ■ permet d'envoyer le dépôt local dans le dépôt distant.
- ☐ permet d'envoyer uniquement les changements des fichiers de travail dans le dépôt local.

Question 14 ♣ La directive ci-dessous

```
#ifndef _CAMION_H_
#define _CAMION_H_
...
#endif
```

- ☐ provoquera une erreur de compilation si elle n'est pas respectée.
- ■ se trouve dans le fichier '.h'.
- ■ sert à éviter les redéfinitions de types/classes/fonctions.
- ☐ se trouve dans le fichier '.cpp'.

Question 15 ♣ Avec Git, quelle commande demande un accès réseau ?
- ■ git pull
- ■ git push
- ☐ git commit

Question 16 ♣ La fonction 'SDL_RenderPresent' de SDL2
- ■ bascule les deux buffers d'affichages.
- ☐ recopie le 2e buffer d'affichage dans le buffer visible.
- ■ met à jour la fenêtre principale en affichant tout les affichages faits depuis l'appel précédent.

Question 17 ♣ Segmentation fault (core dumped) signifie
- ☐ une erreur à la compilation.
- ■ une erreur de segmentation.
- ■ que cette application à causer une erreur due à la mémoire.
- ■ une erreur à l'exécution.
- ☐ une erreur de Doxygen.
- ■ que le système d'exploitation a stoppé cette application.

Question 18 ♣ Avec Git, la commande git pull
- ☐ permet de récupérer uniquement les changements du dépôt distant dans le dépôt local
- ■ permet de récupérer les changements du dépôt distant dans le dépôt local et de mettre à jour les fichiers de travail
- ☐ permet de répercuter uniquement les changements du dépôt local dans les fichiers de travail

Question 19 ♣ Le cahier des charges
- ☐ indique les bureaux où les développeurs seront assis.
- ■ est un contrat entre le client et le prestataire de service.
- ☐ est un cahier à spirales.
- ☐ définit le nombre de niveau qu'aura le jeu.
- ■ indique les modalités d'exécution pour la réalisation avec les éléments à livrer (des démos).
- ☐ propose des cycles accélérés de conception.
- ■ définit exhaustivement les spécifications de base de l'application.

Question 20 ♣ Cette commande 'g++ -ggdb –o projet main.o –L/usr/lib –lABC'
- ☐ '-lABC' indique qu'il faut inclure le ABC.h à la compilation.
- ☐ '-L/usr/lib' indique un chemin vers les fichiers d'entêtes (.h).
- ■ effectue la phase d'édition de lien.
- ■ crée un exécutable en assemblant les fichiers objets.
- ■ '-L/usr/lib' indique un chemin vers les bibliothèques (.a, .so ou .dll).
- ☐ demande la production de fichiers objets (.o).
- ☐ effectue la phase de compilation.
- ■ '-lABC', indique qu'il faut inclure la bibliothèque (library) ABC à l'exécutable.

Question 21 ♣ Avec Git, la commande 'git log'
- ■ affiche l'historique des commentaires entrés lors des commit.
- ☐ enregistre mes changements dans un journal de log.
- ☐ enregistre mes commentaires dans un journal de log.

Question 22 ♣ Avec Git, la commande 'git commit'
- ■ permet d'envoyer uniquement les changements des fichiers de travail dans le dépôt local.
- ☐ permet d'envoyer les changements des fichiers de travail dans le dépôt local et dans le dépôt distant.
- ☐ permet d'envoyer le dépôt local dans le dépôt distant.

Question 23 ♣ Doxygen analyse les commentaires commençant par
- ■ /**
- ☐ /*
- ■ ///
- ☐ /*=

Question 24 ♣ Avec Git, pour récupérer une copie de travail dans le répertoire courant, la commande est
- ☐ git checkout http ://forge.univ-lyon1.fr/monprojet
- ☐ git download http ://forge.univ-lyon1.fr/monprojet
- ■ git clone http ://forge.univ-lyon1.fr/monprojet

Question 25 ♣ Cochez les actions qui sont exécutées classiquement par une boucle d'événement dans une IHM
- ☐ Ouvre une fenêtre.
- ■ Affiche l'application.
- ☐ Sauvegarde toutes les données dans un fichier d'historique.
- ■ Récupère et traite des événements du clavier.
- ■ Récupère et traite des événements de la souris.
- ☐ Pose des questions à l'utilisateur.
- ■ Exécute des actions de manière automatique.
- ☐ Envoie des événements à un autre programme.
- ■ Efface l'écran.

Question 26 ♣ Un 'callback'
- ■ est représenté par un pointeur de fonction.
- ■ est une fonction qui sera appelée par la boucle d'événement.
- ☐ est une fonction qui s'appelle elle-même.

Question 27 ♣ Cochez les balises Doxygen valides.
- ☐ \run pour lancer l'exécution du code.
- ■ \param pour documenter un paramètre de fonction/méthode.
- ■ \fn pour documenter une fonction.
- ■ \brief pour donner une description courte.
- ☐ \ask pour poser une question à l'utilisateur.
- ☐ \save pour sauvegarder toutes les données dans un fichier d'historique.
- ■ \struct \class pour documenter une structure/class C/C++.

Question 28 ♣ 'core dump' signifie que le système a
- [x] sauvegardé tout l'état de la mémoire dans un fichier 'core'.
- [ ] sauvegardé l'historique des instructions dans un fichier 'core'.
- [ ] sauvegardé l'historique des versions de code dans un fichier 'core'.

Question 29 ♣ Avec Git, le dépôt local
- [ ] se trouve sur le serveur Git.
- [x] contient tout l'historique des versions.
- [ ] est l'ensemble des fichiers sur mon disque dur sur lesquels je travaille directement.
- [x] se trouve dans le répertoire '.git'.

Question 30 ♣ Ce code
```
assert(a!=0);
```
- [ ] indique au débogueur de tester si la valeur de a est différente de 0.
- [ ] indique au compilateur de tester si la valeur de a est différente de 0.
- [x] arrête l'application si 'a' vaut 0.
- [x] s'assure à l'exécution que la variable 'a' est différente de 0.
- [ ] indique au débogueur de mettre une valeur différente de 0 dans a.

Question 31 ♣ Cochez les affirmations appartenant à la méthode de conception AGILE
- [ ] Les processus complexes sont importants pour résoudre des problèmes complexes.
- [x] Bâtissez le projet autour de personnes motivées.
- [x] Un logiciel fonctionnel est la meilleure unité de mesure de la progression du projet.
- [x] La méthode la plus efficace pour transmettre l'information est une conversation en face à face.
- [ ] Les processus AGILE promeuvent un rythme de développement tranquille mais sur.
- [ ] On ne revient pas sur un choix quoi qu'il arrive.

Question 32 ♣ Cochez les règles d"écriture de code importantes.
- [x] Variables, fonctions et classes avec des noms claires et explicites.
- [ ] Plus de commentaire que de code.
- [x] Des lignes ne dépassant par 80 colonnes.
- [ ] Le code doit avoir le moins de ligne possible.
- [ ] Toujours indenter avec des espaces et non des tabulations.
- [x] Code indenté.

Question 33 ♣ Cochez les méthodes de conceptions vues durant le cours :
- [ ] Cycle en itératif.
- [x] Cycle en cascade.
- [x] Cycle en spiral.
- [ ] Cycle procédural.
- [ ] Cycle en W.
- [x] Cycle en V.
- [ ] Méthode SOUPLE.
- [x] Méthode AGILE.
- [x] Cycle itératif.

Question 34 ♣ Mon programme ne marche pas, quelle solution ai-je pour le corriger ?
- [ ] De corriger les erreurs de compilation.
- [x] Comprendre ce qu'il fait réellement avec des affichages 'cout/printf'.
- [x] Le tester avec Valgrind.
- [x] Comprendre ce qu'il fait réellement avec un débogueur.

Question 35 ♣ Avec Git, je crée un nouveau fichier 'src/fic.cpp', quelle(s) commande(s) dois-je entrer pour que les autres développeurs puissent récupérer ce fichier ? (ne vous préoccupez pas de l'ordre, cochez toutes les commandes nécessaire).

- ☐ git a src/fic.cpp
- ☐ git add
- ■ git commit -m "ajout de fic" .
- ■ git push
- ☐ git commit
- ☐ git commit -a src/fic.cpp
- ■ git add src/fic.cpp
- ☐ git commit fic.cpp

Question 36 ♣ Ce code

```
0:   class Truc
1:   {
2:      int a;
3:      void faire() const { a=12; }
4:   }
```

- ☐ provoquera une erreur avec un débogueur.
- ☐ provoquera un 'segmentation fault (core dump)'.
- ☐ provoquera une erreur avec Valgrind.
- ■ provoquera une erreur de compilation.