

# TI-Concours 2016

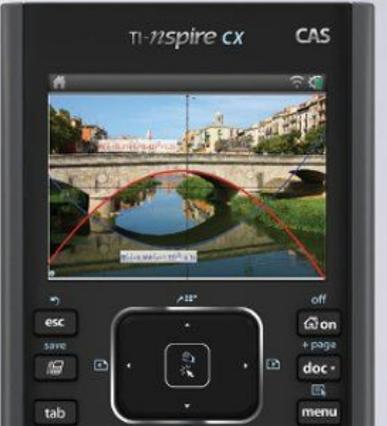
Qualifications, catégorie 1  
(TI-Basic z80 et ez80)

\*\*\*



TI-83 Premium CE

Normal FLOTT AUTO REEL PRG HP  
CALC INTERSECTION  
Y=9.72  
Intersection  
X=1



TI-nspire CX CAS

esc on  
save + page  
tab doc  
menu

TI-Concours 2016

Sur 82/83/84 et Nspire,  
en Basic, ASM, C et Lua

Des calculatrices et  
des goodies à gagner !



TI planet .org

## Avant-propos

Ceci est un sujet de qualification, ce qui signifie que vous avez jusqu'au 08/02/2016 23h59 (UTC+1) pour envoyer votre production à l'adresse [info@tiplanet.org](mailto:info@tiplanet.org). Vous mettez la catégorie (Catégorie 1) en objet, et dans le contenu du message vous indiquerez nom, prénom, et adresse postale complète. N'oubliez pas de joindre votre production, qui sera envoyée sous la forme d'une archive ZIP ou RAR, dont le nom sera votre nom de famille en majuscules sans accents, suivi d'un espace et des caractères « IQ ». Cette archive contiendra l'ensemble des programmes que vous aurez faits (et rien de plus).

Si vous décidez de participer dans les deux catégories, veuillez nous adresser deux courriers électroniques séparés. Vous pouvez mettre à jour votre production autant de fois que vous le souhaitez, en envoyant à chaque fois un courriel comme indiqué ci-dessus. Pour chaque catégorie, nous ne tiendrons compte que de la dernière production reçue dans les délais.

Pour ce sujet, vous devrez pour chaque question écrire un programme, dont le nom sera : les 5 premières lettres de votre nom de famille, toutes attachées et sans accent, suivi des caractères « IQ » et du numéro de la question. Ainsi, si votre nom de famille est De Périgny, pour la question 4 votre programme aura pour nom « DEPERIQ4 ». En procédant ainsi, vous limitez le risque que deux programmes s'appellent pareil. Si malgré cela vous vous rendez compte que quelqu'un d'autre dont les cinq premières lettres du nom sont les mêmes que les vôtres participe, ne vous en souciez pas, nous nous occuperons du problème.

Les programmes doivent tous être écrits intégralement en TI-Basic z80 ou ez80. Cela signifie entre autres que vous ne pouvez utiliser ni librairie externe, ni code assembleur. Nous vous demanderons également de ne pas utiliser de sous-programme, et de ne pas protéger votre code, le but étant tout de même que nous regardions ce que vous avez fait.

Pensez à vous munir d'un câble permettant la connexion entre votre calculatrice et votre ordinateur, et de vous procurer le logiciel de transfert approprié. Si vous avez des difficultés avec ça, n'hésitez pas à poser des questions sur <http://tiplanet.org>.

Sauf mention explicite du contraire, vous n'aurez jamais à demander une saisie de données à l'utilisateur, ni d'afficher quelque chose en sortie. Le sujet vous donnera des noms de variables, vous indiquera ce que chacune d'elles représente, et vos programmes s'exécuteront sous l'hypothèse que ces variables ont été initialisées correctement au préalable. De même, il vous sera toujours spécifié les variables qui devront contenir le résultat demandé. Vous avez le droit de modifier les variables utilisées en entrée via ce que vous programmez, et on suppose qu'il n'y a jamais d'erreur de saisie.

Le barème précis ne sera pas dévoilé, mais sachez qu'environ la moitié de la note portera sur la justesse des résultats produits, et que la seconde moitié portera sur l'efficacité de l'algorithme implémenté. Si vous avez fini longtemps avant la date limite, vous pourrez toujours essayer de rendre vos programmes plus rapides.

La difficulté est globalement croissante, et il n'est pas nécessaire d'avoir tout fait pour vous qualifier pour la suite. Nous vous encourageons à nous soumettre ce que vous avez fait, quel que soit le nombre de questions traitées. Quoi qu'il arrive, nous espérons que vous aurez du plaisir à chercher des réponses !

**Bonne chance à tous !**

## L'évasion

Vous êtes là, assis tranquillement à parcourir l'internet (ou un sujet du TI-Concours, au choix) depuis votre ordinateur. Du moins, c'est la dernière chose dont vous vous rappelez... car après ça, c'est le flou total, et quand vous vous réveillez, vous êtes seul, dans le noir, sans aucun effet personnel sur vous. Alors que la situation vous plonge dans l'incompréhension, vous essayez et ce malgré l'obscurité totale de trouver une sortie.

### Question 1

On suppose que  $[A]$  est une matrice comportant des 0 et des 1, correspondant respectivement à du vide et à des murs. Écrire un programme qui, en supposant que  $[A]$  soit un plan de la pièce où vous vous trouvez, renvoie une variable réelle  $S$  valant 1 si on peut sortir, 0 sinon. On supposera que l'on peut toujours se déplacer librement à l'intérieur d'une pièce donnée, i.e. qu'étant données deux « zones vides » (ou cases valant 0), il existe un chemin dans la pièce pour aller de l'une à l'autre.

Exemple :

[[1,1,1,1,1,1,1]]

Ici,  $S$  vaudra 1, car vous pouvez sortir par la gauche.

[ 1,0,0,0,1,1,1]

S'il n'y avait que des "1" sur la première colonne,  $S$  vaudrait 0.

[ 0,0,1,0,0,0,1]

[ 1,1,1,1,1,1,1]]

\*\*\*

Après avoir longuement tâtonné, vous vous rendez compte que vous êtes entouré de quatre murs, et que la seule porte existante est verrouillée par code secret ! Pour essayer de vous faire entendre, vous tapez sur la porte, puis sur les murs d'à côté. En vain... n'ayant plus rien d'autre à faire, vous vous mettez à taper des combinaisons de chiffres au hasard, en espérant que vous trouviez le bon. Mais, au bout du 42<sup>ème</sup> essai, une voix synthétique, semblant émaner de la porte, vous dit :

« Quel est le plus petit nombre premier strictement supérieur à 9001 ? ».

### Question 2

On suppose que  $N$  est une variable réelle contenant un entier naturel non nul. Écrire un programme qui stocke dans la variable réelle  $P$  le plus petit nombre premier strictement supérieur à  $N$ .

\*\*\*

La porte s'ouvre finalement, et vous rentrez dans la nouvelle salle. Après avoir fait quelques pas, la porte que vous venez d'ouvrir se referme brutalement, et les lumières s'allument au plafond. Vous n'avez aucune issue de sortie ! Au centre de la salle, se trouve un ordinateur. Vous l'allumez, et un document intitulé « Règles du jeu » s'affiche à l'écran.

Dans ce jeu, vous êtes opposé à  $N$  autres participants. Le déroulement s'apparente à celui d'un tableau à élimination directe. A chaque tour, vous rencontrez un adversaire et vous vous battez en duel avec lui. Si

vous gagnez, vous êtes qualifiés et vous disputez la manche suivante, sinon vous êtes mort. Le jeu continue jusqu'à ce qu'il n'y ait plus qu'un survivant.

Chaque joueur (y compris vous) est affecté d'un classement d'entrée, censé être représentatif de votre capacité à gagner ces duels. Le tableau est conçu pour que le meilleur affronte le moins bon, le deuxième l'avant dernier, et ainsi de suite...

Voici quelques exemples de tableaux :

Tableau à deux participants :

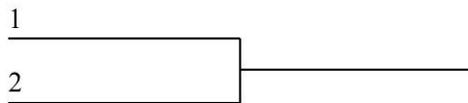


Tableau à quatre participants :

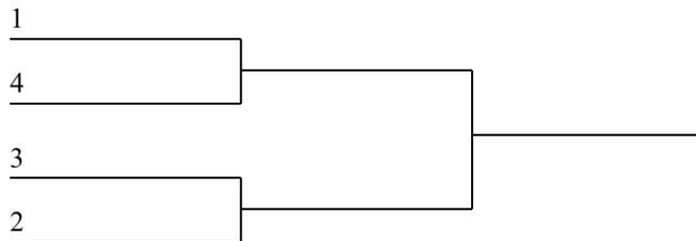
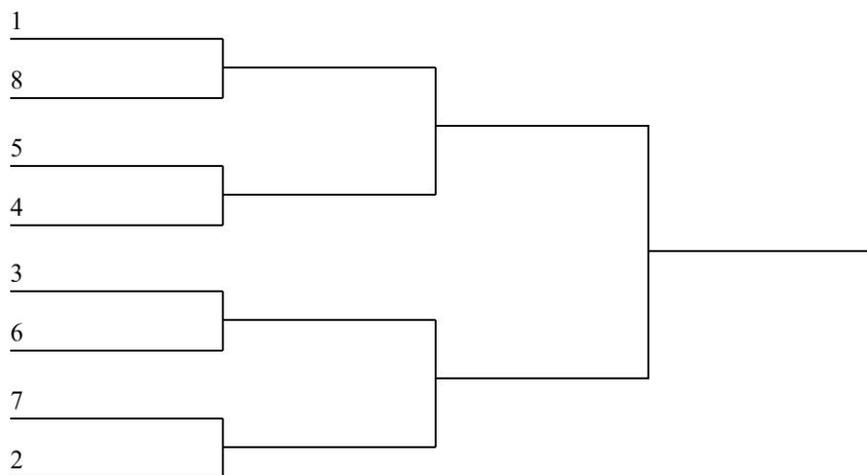


Tableau à huit participants :



On procède de la même manière pour 16, 32, 64... on remarque qu'on a à chaque fois une puissance de 2.

### Question 3

K étant un entier compris entre 1 et 9, écrire un programme qui stocke dans la liste L1 les classements des joueurs qui sont dans le tableau, du haut en bas, au début, dans le cas où ils sont  $2^k$ . Par exemple, pour  $K=3$ , on aura  $L1 = \{1, 8, 5, 4, 3, 6, 7, 2\}$ .

\*\*\*

### Question 4

Étant donné N le nombre de participants et C votre classement au début du jeu, écrire un programme qui, toujours dans l'hypothèse où N est une puissance de 2, stocke dans la variable réelle A le classement de votre premier adversaire.

\*\*\*

Bien évidemment, le nombre de joueurs n'est pas toujours une puissance de 2. Les meilleurs sont alors exemptés du premier tour (également appelé « tour préliminaire » ou « tour de barrage »). On peut se convaincre qu'il existe alors un et un seul tableau à élimination directe qui d'une part vérifie la définition donnée ci-dessus, et qui d'autre part ne permet pas à tous les joueurs d'être exemptés du premier tour. Par exemple, s'il y a 7 joueurs, alors le 1<sup>er</sup> est exempté des quarts de finale, tandis que le 5<sup>ème</sup> affronte le 4<sup>ème</sup>, le 3<sup>ème</sup> le 6<sup>ème</sup>, et le 7<sup>ème</sup> le 2<sup>ème</sup>. Ainsi, le premier duel du 1<sup>er</sup> l'opposera au vainqueur du duel entre le 5<sup>ème</sup> et le 4<sup>ème</sup>.

### Question 5

Étant donné N le nombre de participants et C votre classement au début du jeu, écrire un programme qui stocke dans la liste réelle L1 les classements de vos potentiels adversaires pour votre premier duel.

\*\*\*

Tous vos adversaires étant munis d'une intelligence artificielle, il vous sera impossible de vaincre quiconque sans une quelconque aide extérieure. Fort heureusement, l'ordinateur mis à votre disposition dispose d'une base de données recensant toutes les intelligences artificielles que vous êtes susceptibles de rencontrer, ainsi que leurs forces et leurs faiblesses. Ainsi, en déterminant à l'avance vos futurs adversaires, vous avez une chance de vous en sortir en exploitant leurs points faibles, que vous aurez étudiés à l'avance !

On suppose désormais que les duels sont organisés tour par tour, de haut en bas dans le tableau. Ainsi, dans le cas où  $N=7$ , les duels seront numérotés comme suit.

Numéro du duel	Adversaires
1	Le 5ème contre le 4ème
2	Le 3ème contre le 6ème
3	Le 7ème contre le 2ème
4	Le 1er contre le vainqueur du duel 1

5	Le vainqueur du duel 2 contre le vainqueur du duel 3
6	Les deux finalistes

## Question 6

On suppose que L1 contient la liste des résultats des précédents duels : le terme numéro  $i$  vaut 1 si c'est l'adversaire le plus en haut dans le tableau qui gagne, et vaut 2 sinon.  $N$  et  $C$  étant toujours égaux respectivement au nombre de participants et à votre classement, écrire un programme qui stocke dans L2 la liste des classements de vos adversaires potentiels au prochain tour. Si vous êtes déjà mort, L2 sera égal à  $\{-1\}$ .

\*\*\*

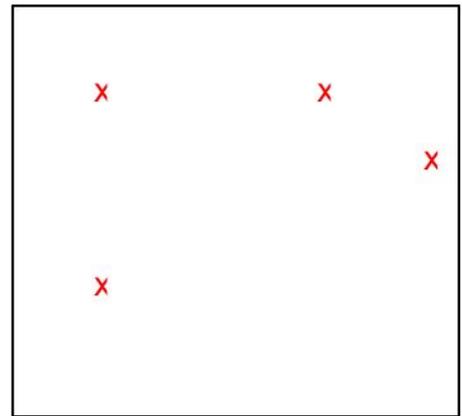
Le jeu débute. En appliquant la stratégie élaborée précédemment, vous venez à bout de tous vos adversaires. Une fois le duel final remporté, une porte non verrouillée apparaît dans le mur. Au moment où vous l'ouvrez, la lumière s'éteint... vous avancez à l'aveugle dans la nouvelle pièce. Et là, encore une fois, la porte se ferme brusquement derrière vous et les lampes au plafond s'allument. Et là – horreur ! – la pièce est remplie de zombies ! En vous voyant, toutes ces infâmes créatures vous foncent dessus. Alors que vous vous demandez si votre fin est arrivée, vous remarquez une petite porte à votre droite. Saisissant votre chance, vous foncez vers ce qui semble être votre salut, et vous vous engouffrez dans l'ouverture.

Après avoir bloqué la porte (qui cette fois possédait un loquet), vous vous retrouvez face à des escaliers. Vous les gravissez et vous vous retrouvez alors dans une sorte de salle de contrôle, avec des fenêtres. En vous approchant, vous réalisez que ces dernières donnent sur la salle dans laquelle vous vous trouviez un instant auparavant. En portant votre regard sur le reste de la pièce, vous remarquez que la salle infestée de zombies dispose d'un système de défense en état de fonctionnement. Vous disposez au plafond d'un grand nombre de vaporisateurs permettant de désintégrer l'ennemi sur une zone dans laquelle vous définissez vous-même l'étendue et la position. Néanmoins, plusieurs contraintes vous empêchent de faire ce que vous souhaitez :

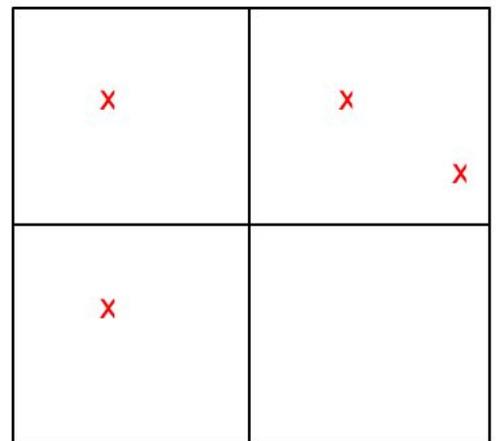
- le temps de mise en route étant long et bruyant, si toute la pièce n'est pas visée par un vaporisateur en même temps les zombies pourront esquiver vos attaques;
- la désintégration n'opère pas si un vaporisateur vise une zone présentant deux zombies ou plus ;
- si deux vaporisateurs visent une même zone, cela provoque un bogue fatal mettant hors-service à jamais le système de défense et vous condamnant à mourir de faim dans votre salle de contrôle.

La pièce étant carrée, vous décidez d'adopter la stratégie suivante : s'il y a strictement plus qu'un zombie, vous coupez la salle en quatre (dans votre tête, bien sûr). Pour chaque zone de la salle encore « intègre », vous refaites un nouveau découpage si la condition qui vient d'être évoquée n'est toujours pas respectée. Et ainsi de suite...

Voici un exemple. Les zombies sont disposés dans la salle comme suit.

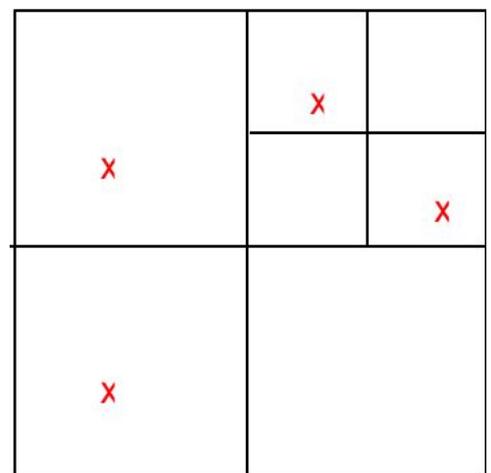


Il y a plusieurs zombies, donc vous découpez en quatre.



Dans la zone en haut à droite il y a encore deux zombies, donc on recommence, uniquement dans cette zone.

Ainsi, vous définissez sept zones distinctes contenant au plus un zombie, ne se chevauchant pas, et formant une partition de la salle. Cela signifie que si pour chaque zone vous utilisez un vaporisateur, et que vous lancez tout les processus en même temps, vous pourrez arriver à vos fins.



## Question 7

On suppose que la salle est un carré de côté  $C$ , et que les coordonnées du sommet en bas à gauche sont  $(0,0)$  dans un repère orthonormé arbitraire. Écrire un programme qui à partir de deux listes  $L1$  et  $L2$  représentant les abscisses et les ordonnées des zombies (le zombie  $i$  a pour coordonnées  $(L1(i), L2(i))$ ), stocke dans la variable réelle  $N$  le nombre de vaporisateurs utilisés par la méthode qui vient d'être décrite. Avec l'exemple précédent, on a  $N=7$ . On suppose qu'on a toujours assez de vaporisateurs à notre disposition, et qu'il n'y aura jamais de zombie situé pile sur une ligne.

\*\*\*

Une fois les zombies désintégrés, pendant un long moment rien ne se passe. C'est alors qu'une porte apparaît dans le mur de la salle. Vous ouvrez la porte, et encore une fois, tout devient noir. Vous avancez, et là... vous ne savez plus ce qui s'est passé. Ce qui est sûr, c'est que vous êtes à nouveau dans votre chambre, devant votre ordinateur, et qu'il est 7 heures du matin. Peut-être n'était-ce qu'un mauvais rêve ?

FIN DU SUJET

\*\*\*