NSI (Numérique et Science de L’informatique)

DST d’NSI

Appréciations :

Note :

\_\_\_\_\_\_\_\_\_\_\_\_

Signature :

Problème 1 :

PARTIE A :

1. Curseur = cur

Séquence e1S2S2Ge2 ;

|  |  |  |
| --- | --- | --- |
| **Détaille de ce que fait la machine quand elle lit un « blanc »** | | |
| **Symbole** | **signification** | **action effectuée** |
| e1 | E1=Etat | le cur se place à l’état 1 |
| S2 | Banc | il lit S2 donc marque un blanc |
| S2 | Blanc | idem |
| G | Gauche | il lit Gauche donc se déplace vers la gauche |
| e2 | E2 | il lit E2 donc se place à l’état 2 |
| ; | Fin | il lit Fin donc s’arrête |

Pour cette suite, prenons le cas que la tête de lecture, si elle Lit S0 (soit 0 en binaire) elle écrit un 1, si elle lit S1 (soit 1 en binaire) elle écrit un 0, si elle lit S2 (soit blanc) elle exécute la séquence e1S2S2Ge2 ; (décrite dans le tableau « Détaille de ce que fait la machine quand elle lit un « blanc » »).

Cur : ↓

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | S2 | S1 | S1 | S0 | S1 | S2 | S2 |

La machine lit un 1, donc elle écrit un 0

Cur : ↓

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | S2 | S0 | S1 | S0 | S1 | S2 | S2 |  |

La machine lit un autre 1, donc elle écrit un 0 à nouveau

Cur : ↓

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | S2 | S0 | S0 | S0 | S1 | S2 | S2 |  |  |

La machine lit un 0, donc elle écrit un 1

Cur : ↓

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | S2 | S0 | S0 | S1 | S1 | S2 | S2 |  |  |  |

La machine lit un 1, donc elle écrit un 0

Cur : ↓

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | S2 | S0 | S0 | S1 | S0 | ; |  |  |  |  |  |

La machine lit un ; , donc elle s’arrête.

Pour ei=Di\*A, D=i\*C, ;=e1S2S2Ge2

Calcule final de la machine de Turing : S2S0S0S1S0 ;

S1=DA

S2=DCC

G=G

AA=e2

DADCCDCCGAA ;

S2S0S0S1S0 ; transformé donnera

DCCS0S0DAS0

Si A=1, C=2, D=3, G=4, R=5, ;=7, S0=0

DADCCDCCGAA ; DCCS0S0DAS0

3 1 3 2 2 3 2 2 4 1 1 7 3 2 2 0 0 3 1 0

Problème 2 :

PARTIE A :

|  |
| --- |
| def puissance(x,n):  if n==0:  x=1  return x  elif n>0:  x=x\*(x\*puissance(x,n-1))  return x  print(puissance(10,100)) |

PARTIE B :



|  |
| --- |
| from random import \*  def isertion(list, x, n):  if n>0:  list[n-1]=x  return list, x, n  else: isertion(list, list[randint(0,len(list))], randint(0,len(list)))  list=[]  for i in range(randint(0,randint(100,200))):list.append(randint(randint(-200,0),randint(100,200)))  x=list[randint(0,len(list))]  n=randint(0,len(list))  isertion(list, x, n)  print(list) |

1. Pas réussi

|  |
| --- |
| from random import\*  def tri\_insertion(list,liste\_triee):  for i in list:  elementATrier=i  for I in range(len(liste\_triee)):  if elementATrier<list[I]:  liste\_triee[I].append(elementATrier)  elif elementATrier==list[I]:  liste\_triee[I].append(elementATrier)  elif elementATrier>list[I]:  liste\_triee[I+1].append(elementATrier)  return liste\_triee  liste\_triee=[]  list=[]  for i in range(randint(0,randint(100,200))):list.append(randint(randint(-200,0),randint(100,200)))  print(tri\_insertion(list,liste\_triee)) |

PARTIE C :

|  |  |
| --- | --- |
| Non Récursif | Récursif |
| from turtle import \*  couleurs=['blue','green','yellow','orange','red','purple']  bgcolor('black')  def dessin():  for i in range(180):  color(couleurs[i%6])  forward(i)  right(59)  dessin(tours) | from turtle import \*  couleurs=['blue','green','yellow','orange','red','purple']  bgcolor('black')  def dessin(tours):  if tours!=180:  color(couleurs[tours%6])  forward(tours)  right(59)  dessin(tours+1)  else:  return tours  tours=0  dessin(tours) |