import random  
  
input\_file = *open*("23341065\_Istiaque\_CSE422\_08\_Lab\_Assignment02\_InputFile\_Summer2024.txt")  
output\_file = *open*("23341065\_Istiaque\_CSE422\_08\_Lab\_Assignment02\_OutputFile\_Summer2024.txt", "w")  
n, t = *map*(*int*, input\_file.readline().split())  
courses = [input\_file.readline()[:-1] for i in *range*(n)]  
  
  
def fitness(*s*):  
 overlap\_penalty = 0  
 for i in *range*(t):  
 cnt = 0  
 for j in *range*(n):  
 cnt += (*s*[i \* n + j] == '1')  
 overlap\_penalty += *max*(0, cnt - 1)  
  
 consistency\_penalty = 0  
 for i in *range*(n):  
 cnt = 0  
 for j in *range*(t):  
 cnt += (*s*[j \* t + i] == '1')  
 consistency\_penalty += *abs*(cnt - 1)  
 return -overlap\_penalty - consistency\_penalty  
  
  
def selection(*n*):  
 return population[random.randint(0, *n* - 1)], population[random.randint(0, *n* - 1)]  
  
  
def crossover(*parent1*, *parent2*):  
 return *parent1*[:*len*(*parent1*) // 2] + *parent2*[*len*(*parent1*) // 2:]  
  
  
def mutate(*s*):  
 id = random.randint(0, *len*(*s*) - 1)  
 return *s*[:id] + random.choice("01") + *s*[id + 1:]  
  
  
population = [''.join(random.choices("01", *k*=n \* t)) for i in *range*(10)]  
tmp = population *# soterd initial population for later part*population.sort(*key*=lambda i: -fitness(i))  
  
*# part 1*for i in *range*(10):  
 if fitness(population[0]) == 0:  
 break  
 next\_gen = []  
 for j in *range*(10):  
 parent1, parent2 = selection(*len*(population))  
 offspring1, offspring2 = crossover(parent1, parent2), crossover(parent2, parent1)  
 next\_gen.append(mutate(offspring1))  
 next\_gen.append(mutate(offspring2))  
 next\_gen.sort(*key*=lambda i: -fitness(i))  
 population = next\_gen  
output\_file.write(f"{population[0]}\n{fitness(population[0])}\n")  
  
  
*# part 2*def two\_point\_crossover(*parent1*, *parent2*):  
 point1 = random.randint(1, *len*(*parent1*) - 2)  
 point2 = random.randint(point1, *len*(*parent1*) - 1)  
 return (*parent1*[:point1] + *parent2*[point1:point2] + *parent1*[point2:],  
 *parent2*[:point1] + *parent1*[point1:point2] + *parent2*[point2:])  
  
  
parent1, parent2 = random.choices(tmp, *k*=2)  
offspring1, offspring2 = two\_point\_crossover(parent1, parent2)  
output\_file.write(f"Parent 1: {parent1}\nParent 2: {parent2}\nOffspring 1: {offspring1}\nOffspring 2: {offspring2}\n")  
  
input\_file.close()  
output\_file.close()