

EXPERIMENT 07

CODE:

```
def minimax(numbers, is_maximizing): if not numbers: return 0, None
if is_maximizing: pick_start_score, _ =
minimax(numbers[1:], False) pick_start_score +=
numbers[0]

pick_end_score, _ = minimax(numbers[:-1], False)
pick_end_score += numbers[-1]

if pick_start_score > pick_end_score:
return pick_start_score, 'start' else:
return pick_end_score, 'end' else:
pick_start_score, _ = minimax(numbers[1:], True)
pick_end_score, _ = minimax(numbers[:-1], True)

if pick_start_score < pick_end_score:
return pick_start_score, 'start' else:
return pick_end_score, 'end'
numbers = [3, 5, 2]

best_score, best_move = minimax(numbers, True) print(f"Best score the maximizer can achieve:
{best_score}") print(f"Best move at first step: pick from the {best_move}")
```

OUTPUT:

```
PS C:\Users\computer1\Desktop> & C:/Users/computer1/anaconda3/python.exe c:/Users/computer1/Desktop/Untitled-1.py
Best score the maximizer can achieve: 5
Best move at first step: pick from the end
PS C:\Users\computer1\Desktop>
```

EXPERIMENT 08

CODE:

```
add(X, Y, Z) :-  
Z is X + Y.
```

OUTPUT:

```
GNU Prolog 1.5.0 (64 bits)  
Compiled Jul  8 2021, 12:22:53 with gcc  
Copyright (C) 1999-2021 Daniel Diaz  
  
| ?- consult('C:/Users/computer1/Desktop/add.pl').  
compiling C:/Users/computer1/Desktop/add.pl for byte code...  
C:/Users/computer1/Desktop/add.pl compiled, 1 lines read - 377 bytes written, 2 ms  
  
yes  
| ?- add(4, 6, Result).  
  
Result = 10  
  
yes  
| ?- |
```