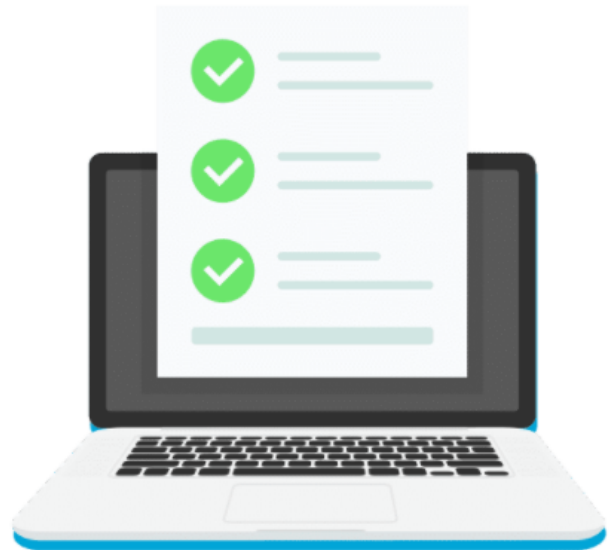




CS354

Assignment-3



Somya Mehta
190001058



Q1:-

Code:-

% idea:- we replace the head with the desired element

% if head comes to be the variable which we want to replace

% otherwise we move ahead by recursion

```
replace([],X,Y,[]).
```

```
replace([H|T],X,Y,[Y|List]):-
```

```
    H==X,
```

```
    replace(T,X,Y,List),!.
```

```
replace([H|T],X,Y,[H|List]):-
```

```
    replace(T,X,Y,List).
```

Screenshot:-

```
/home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q1.pl:3: warning: sing
/home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q1.pl compiled, 7 line

(1 ms) yes
| ?- replace([1, 2, 3, 4, 3, 5, 6, 3], 3, x, List).

List = [1,2,x,4,x,5,6,x]

yes
```



Q2:-

Code:-

% idea:- we recurse through and find $f(n-1, X)$ & $f(n-2, Y)$ to calculate $f(n, Z)$

fibonacci(0,1):-!.%base case when 0-th number is asked

fibonacci(1,1):-!.%base case when 1-st number is asked

fibonacci(X,Ans):-

$X \geq 0$,

 Z is $X-1$,

 Y is $X-2$,

 fibonacci(Z,Ans1),fibonacci(Y,Ans2),

 Ans is $\text{Ans1} + \text{Ans2}$.% arithmetic operation



Screenshot:-

```
| ?- ['Q2.pl'].
compiling /home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q2.pl for byte code...
/home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q2.pl compiled, 7 lines read - 1242

(1 ms) yes
| ?- fibonacci(1,X).

X = 1

yes
| ?- fibonacci(3,X).

X = 3

yes
| ?- fibonacci(5,X).

X = 8

yes
| ?- fibonacci(6,X).

X = 13

yes
```

Q3:-

Code:-

% idea:- we insert the value at the start with the desired element

% if we have to insert at 1st position

% otherwise we move ahead by recursion and take the head into our final list

%and then insert the element at k-1 th position in the remaining list



`insert_at(X,L,1,[X|L]):-!.%Base case when we have to insert at 1st position`

`insert_at(X,[H|T],K,[H|List]):-`

`KK is K-1,`

`insert_at(X,T,KK,List).`

Screenshot:-

```
| ?- ['Q3.pl'].
compiling /home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q3.pl for byte code..
/home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q3.pl compiled, 3 lines read -

yes
| ?- insert_at(alfa,[a,b,c,d],2,L).

L = [a,alfa,b,c,d]

yes
```

Q4:-

Code:-

`% Similar to insert_at function here also we recurse in the similar way i.e`

`%if we have to remove the 1st element from the list then we simply add a tail to our answer`

`%for other cases we take Head and then remove the (k-1)th element from the remaining list`

`remove_at(H,[H|T],1,T):-!.%base case`

`remove_at(X,[H|T],K,[H|L]):-`

`KK is K-1,`

`remove_at(X,T,KK,L).`



Screenshot:-

```
| ?- ['Q4.pl'].
compiling /home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q4.pl for byte code...
/home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q4.pl compiled, 3 lines read - 925 bytes

(2 ms) yes
| ?- remove_at(X,[a,b,c,d],2,R).

R = [a,c,d]
X = b

yes
```

Q5:-

Code:-

% idea take each start element and then recurse the range function from start +1

range(Start,Start,[Start]):-!.%base case when we have only 1 element in our range

range(Start,End,[Start|L]):-

End>=Start,

L is Start+1,

range(L,End,L).



Screenshot:-

```
| ?- ['Q5.pl'].
compiling /home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q5.pl for byte code...
/home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q5.pl compiled, 5 lines read - 8

yes
| ?- range(4,9,X).

X = [4,5,6,7,8,9]

yes
```

Q6:-

Code:-

%idea:- gcd(a,b) so when $a > b$ then we divide a by b and see the remainder

% then our answer would be gcd(b,a%b). Similarly we can handle the case when $a \leq b$

gcd(L,0,L):-!.%base cases i.e when either of the number is 0

gcd(0,L,L):-!.%then the gcd would be the other number which is not 0

gcd(L,S,Ans):-

 L>=S,!,

 Rem is L mod S,

 gcd(S,Rem,Ans).

gcd(L,S,Ans):-

 L<S,!,

 Rem is S mod L,

 gcd(L,Rem,Ans).



Screenshot:-

```
| ?- ['Q6.pl'].
compiling /home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q6.pl for byte code...
/home/somyamehta_24/sem6/CI/Prolog/Assignment3/Q6.pl compiled, 9 lines read - 1472 bytes written

yes
| ?- gcd(7,27,X).

X = 1

yes
| ?- gcd(57,27,X).

X = 3

yes
| ?- gcd(1,30,X).

X = 1

yes
| ?- gcd(56,28,X).

X = 28

yes
```