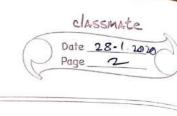


FUNCTIONAL PROCT. IN COQ

	(semple v
Inductive day: Type :=	
monday	100000
170	
1 tuesday	
wednesday	
1 thorsday	New York
1 friday	Survey k
1 saturday	. 38) .
1 Sunday.	
- Jose Jose of	Topingor To
Definition next-weekday (d:	
match d with	
1 monday >> tuesday	4,5
I tuesday & widnesday	
wednesday > thursday	
1 thorsday & friday	
friday > monday	
Saturday >> monday	, de 30f
Sunday >> monday	1 3 3 d by
end.	619
* beef below : beef :	1 des northers
compute (next-weekday fri	day)
monday day	- July 1
	1/ 1/1/1/1
ompute Chept weekday Chext	weekday Saturday).
1 tuesday: day	



Example test next-weekday?

(next-weekday friday (next-weekday so brobay) = twog

Proof simple reflexivity. Qed.

Inductive book: Type :=

three

| false.

Definition negb (b:book):book :=

I true => false Arios be determined

I false => true justisent <= justinom

end. just sombout <= justisent

Definition and b (b) bis bood (b) (b) bis bood :=

match bi without nom <= justinom

I false >> false nom <= justinom

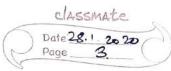
end.

1 false >> false nom <= justinom

end.

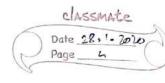
| true > true | the parties of the p

Definition orb (b: bool) (62: bool) !bool :=



Example test_orbl: (orb true false) = true
Proof. Simpl. reflexivity. Qed. Example test-orbz: (orb folse false) = false Proof simpl. reflexifity. Ged. Example test_orbs: (orb folse true) = true. Proof. simpl. reflexificity. Qed. Example test orby: (orb true true) = true. Proof. simpl. reflexivity. and. Notation "x BB y" = (andb ay). Motation "x 11 g" := (orbornely) Example test-orb 5: false Il balse Il true Proof. simpl. reflexivity. Qed. NOTE: simple factic won't expand multi-level bunction calls. for that we may use unfold ign). Definition hands (5. 16001) (62 : 6001) : bool := match 61 with true => (*(negb b2) +

end.



Proof. reflerivity. Qual. Frample test_norndb2: (nandb folse folse) = true.
Proof. reflexivity. Qed. Example test_nandb3: (nandb false true) = true.
Proof. reflexivity. Qed. Example test_nandbh: (nandb true true) = false.
Proof. reflexitivity. Qed: Definition andba (bi: book) (bz: book): book := l true => (and b b2 b3) ent 11 sold false of 1302 false: 2 dro-test signari Deck simple vellex, vity, acd. ··· examples. single lacke want expend multi-less LACTION CALLS. for that we man 239Y Turple Col Check true. (see => Strue : book it) about not (heck (negb true).



Check negb. (+ ===> negb: bool -> bool +) NEW TYPES FROM OLD Inductive rgb: Type := l green Inductive color: Type := =: L white (sldayn : dn) 2125. Ll nor I primary (p: rgb) Definition monochrome (CC: colon) : bool := match c with I black as true a sud White 3 true (+ soon of the 1 primary q => folse · ==> +ne: boot +) Definition is red (c: color): bool := match c with I black => false

I white > bake 1 primary - > false end.

	CIAS	ssmate	
	Date 2	e. cron	5/
1	Page_	6.	(

TUPLES.

1 B. .

Inductive bit: Type:=

Inductive nybble: Type:=

| buts (bo b) b2 b3: bit).

Check (bits B, Bo B, Bo).

(* ===> bits B, Bo B, Bo : nybble *)

= copy and she hat

Definition all-zero (nb: nybble) : book :=

1 Chits Bo Bo Bo Bo Bo) >> true
1 Chits - - - >> false

Dol Miller monschieme (Conceller) - bus :-

Compute Call-zeno Chits B, B, B, B,))

11(===> false : book *)

Compute (all-zero (bits Bo Bo Bo Bo)).

(* ===> true: boo(*)

with is red (c. aler): beach

. 10 > 10 (cc

Modele Nat Playground.

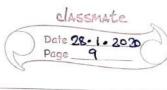
36-1 6

NUMBERS Inductive vat : Type := 1 s Cn: (nat). Inductive nat': Type := 1 stop 1 tick (boo = nat!). Definition pred (n: nat): nat := match n with 1 Sn1 > n1 end. =:)ood : (+op on) Ends NotPlayground. Check (s (s (s o)))). (* ===> 4: nat *) eddle (hingt): book :=

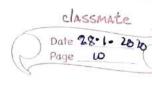
Definition minoster	o Cn:n	at) : n	at := 11
match n with			
10 00			· 10.j
180 30			0
1 (s (s n') >	n'	, ,	
end.			
Cius			
Compute (minustivo	4).		- b. L
(* ===> 2: nat)			
			11.1

Check predo			
			0.1
Fixpoint evenb. (n			
	: Mar J	. 2006 .	- 1)
match n with			
0 >> true			
180 >> false	rd;	-3/11/20	1-0/ 10-
SO >> false S(S ni) >> (eve	nb n')		
end.			
(77		6 23
•		* ton .	

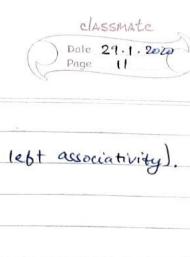
Definition oddb (h: not): bool:=
Chego (evenb n)).



Example test oddb1: oddb 1 = true. Proof. Simple reflexivity. Qed. Example test_oddb2: oddb 4 = false. Proof simpl - reflexivity. Qed. Module Nat Playground 2. house grant toll but Fixpoint plus (n: nat) (m: nat) : nat :== match n with 0 3 m 02 0 (| Sn' >> S (plus n'm) end. Compute (plus 3 2). Fixpoint mult (n m : nat) : nat := match n with 0 7 0 I sn' > Come (plus m (mult n' m)) end. Example test-molti: (mult 3 3) = 9. Proof - simpl. reflexivity Qed



Fixpoint minus (n m: nat) : nat := match n, m with 10,- >0 1s_, 0 > n Isn's smi > (minus n' mi) end. End Nat Playground? Fixpoint expto (base power nat) : nat := match power with alms a 15tow 0 > 80 1 Sp 3 (mult base (exp base p)) end. (- por (plus 3 2). Fixpoint factorial (n: nat): nat := match n with = count mult (n m : nat) : 008 (= 0 1 I Sn' > (mutt n (factorial n'))toni end.
(m'n +6m) m 2019) word) < 'n2 | Example test-factorial : (factorial 3) = 6. Proof. reflexifyinged: How tot some First simple reflexivity and Example test-bactorial2: (bactorial 5) = (mult 10 12) Proof reflexivity. Red.



Motation "x+y" := (plus x y) (at level 50, left associativity). : nat-scope. Notation "n-y" := (minus x y) (at level 50, left associativity)

· nat-scope Notation "xxy" := (multiny) (at level 40, left associativity) : nat-scope.

Check ((0+1)+1). ma) del aviting Treats (leteral) and

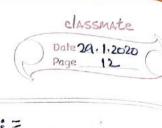
-2101 < JUN 1 Fixpoint eqb (n m: nat) ; bool ==1

match n with & SUVF 10 => match m with 10 of true Jan

end.

|S-=> balse | Sni => match m with 10 => false

- Ismi > (eqb n'm') endo



Fixpoint (eb (n m: nat): nat := match n with 10 => match my with true 0 > losse true I Sm > match m with 1. 0 >> bake Ismi > (leb n' m') end problendal on love in Definition (to (n m : nait) match (leb n m) with | false > false I true => match (leb m n) with true > balse I false & strues end over coi end. match or ever autor C of Theorem plus-o-n: boralln: nat, 0+n=n. Proof. . contras n. Simpl. reflexitivity. Qed.