


I'm not robot  reCAPTCHA

I'm not robot!

Solomon keep haxxor stuff code

Reed-solomon code solved example.



Common reed solomon codes. Solomon reed code.

Buyable start skills: In this section, you need the following:- 1: Lightning- 2: Fire Ball- 3: Magic Shield or Battlemage- 4: Stun or Chaining This might take a while. A perfect mage can be done by 1, 2 and a variation of 3. Gear: The gear suggested can be obtained on Archmage difficulty or higher (Except the experience gear). My suggestion is that you damage a boss to a low percentage of health and then use your Magic Key to teleport back to the town. (This allows you to safe) You can now repeatedly die and rekill the boss until the loots suits your needs. OBS: Keep min. 1 room alive for you to die as if you enter a new floor you will save the game. Gold: Attempt to get a 1000% or 500% gold staff on a random character. Use this character to a high amount of gold. Never sell any item you gain and just before you enter a new difficulty, buy every item available at the item shop. You will, at the end of Archmage or whatever, have an insane amount of items, which you can trade to a potential perfect mage. Items can be traded by the Amazing Cabinet next to the item shop. The perfect mage: Grind a lot of gold to a character that has perfect buyable start skills. Buy Lightning max immediately and grind 1st floor with experience gear on. You will, hopefully, get either Fireball, Magic Shield, Chaining or Stun right away. Whenever you get a new skill, which is buyable, max it. If you do not have enough gold - go grind it. Eventually, you will have full lightning, full fireball, which it what matters. When you do, each time you are about to level, equip full stats gear (+7 all stats). You will, eventually, be able to take Flame Lash, but only do so, if you are full in lightning and fireball with stats gear on. The reason being is that it will merge a 27 lightning ball with a 27 fireball for insane dmg, even when the +7 gear is not equipped. Moreover, you will chain +7 targets. Your mage is pretty close to perfect now. All that is left in order to maximize it is to get Siege Mage and Faster Caster up, everytime you level. Leveling: Leveling is no mystery - Always equip full experience gear until Gauntlet, no matter how often you die. One note regarding this is, that you will level on Solomon no matter what amount of experience you have gathered in that level - take advantage! This means that you should always have a lot of experience to go before you kill Solomon - killing him on 90% experience is just foolish. How can I avoid this? Buy max experience just before you kill last pack before Solomon. iPhone - iPad Games Home / Sub Menu Developer: Raptisoft Publisher: Raptisoft Genre: Role-Playing Release: Apr 3, 2010 Platform: iPhone - iPad ESRB: Not Set Advertisements Codes Req: Notepad++ , jailbreak iphone, winscp or iphone browser! /user/applications/*-*.*/sk.app/data/ in that folder is all the items, magic's scripts its plain text you can edit! Item Exampel (this will add 250 more dmg and make all skills go up 200) STAFF Effect Updamage: +250 Effect UpAllSkills: +200 \$1 Spell Exampel (from low dmg to high dmg) dmg on skill level 0 DAMAGE1_0=1.5 we change to DAMAGE1_0=60.5 it will pwn all :) Hints Tip 1: Track your stats to know when you are just about to level up. Use the town portal door to return to town then go back to where you left off. If you do not like the random skills assigned when you level up, intentionally die. You will restart at the portal door again. Repeat the process until the upgrade skills offered to you upon leveling up are acceptable. Tip 2: When you level up and you do not like the skills offered, press [F0d] to shut down the game. When the game is resumed, the level up screen will immediately reappear, but with new random skill choices. Tip 3: In Hardcore mode, the game ends when you die. It also becomes increasingly difficult to stay alive as you go into harder difficulties. When you die, your wizard freezes in place then disappears and your hat drifts down.



If you exit the app and shut it off while your hat is falling or just after you run out of health, when you open the app again you will be alive and well at the start of that floor. Glitches Enter town and be in town in the save with the item(s) you want duplicate. In your current save sell an item, potion, etc. Exit the save and go to the save with the item(s) you want to duplicate. Place them in the magic cabinet. Sell another item or something. This should crash the game after a moment. Also, be sure to have completely left the game and enter again once you are in town. You can sell ANYTHING when you are selling stuff. The items can be sold in any save and begin the process again. This works the other way also. If you begin the process with items in the cabinet already, pull them out, and sell the second item they WILL be deleted. This works because it resets your inventory but not the cabinet when it crashes to where it started that save. You can buy items while in the first part. You don't have to use 2 different saves, just exit and re-enter a save. DO NOT FULLY EXIT THE GAME! It will crash but that's what you need to happen. I don't know why this happens, I've only tested this on an iPhone 3GS with ios 6. I don't know if this works with extra cabinet slots either... Submit Cheats If you know cheat codes, secrets, hints, glitches or other level guides for this game that can help others leveling up, then please Submit your Cheats and share your insights and experience with other gamers. Latest Questions Ask a Question Stuck in this game? Ask a question below and let other gamers answer your question or view answers to previously asked questions. If you think you are an expert then please try to help others with their questions. Your Question Gameplay Videos Solomon's Keep - Gameplay Video More Options Need Help? Ask A Question Here! Advertisements Games You May Like Haxxor is a simple encryption/decryption framework, which support multiple methods of encryption. More methods will be added as time goes on. The hashes that are generated by Haxxor are all returned as a base-64 string. In some cases when a method is able to support decryption, the key and IV used to encrypt will also be included within the returned hash, such as with AES in the format "key:iv:cipher". This is done so later decryption is possible. Haxxor is more meant as a tool to demo encryption/decryption, and is not meant to be 100% secure. Haxxor is however open-source, so you are able to modify the underlying code to make it more secure. Features Support for multiple methods of encryption and decryption Simple factory class to choose a method Ability to cycle a cipher through multiple methods of decryption when method is unknown Can check if some plain text matches a passed cipher Comes with a console application Comes with a simple real-time encryption/decryption GUI SHA 1, 256, 384, 512 AES 128, 256 MD5 RIPEMD 160 More to come soon. Examples To use the GUI, open up the Haxxor GUI application. Console > Haxxor.Console encrypt aes128 "Hello, world!" > Haxxor.Console decrypt aes128 "hash" > Haxxor.Console list > Haxxor.Console help Code To use the framework, reference the Haxxor.Framework in your project. // Retrieve a module for AES128 encryption var module = HaxxorFactory.GetByType(EncryptionType.AES128); // Retrieve a module by hash // (this is determined by the tag that Haxxor places on a hash, ie "AES128") var hash = HaxxorFactory.GetByType(EncryptionType.AES128); var hash = module.Encrypt("Hello, world!"); module = HaxxorFactory.GetByHash(hash); Console.WriteLine(module.Decrypt(hash)); // Cycle a cipher through all possible decryption methods, if the original method is unknown var attempts = HaxxorFactory.Cycle("some hash here"); // Each module also has a property informing if the method supports decryption var module = HaxxorFactory.GetByType(EncryptionType.SHA1); Console.WriteLine("Do I support decryption?" + module.IsDecryptable); //Finally, you can validate some plain text matches a cipher var module = HaxxorFactory.GetByType(EncryptionType.AES128); var hash = module.Encrypt("Hello, world!"); var matches = module.Validate("Hello, world!", hash); Page 2 Haxxor is a simple encryption/decryption framework, which support multiple methods of encryption. More methods will be added as time goes on. The hashes that are generated by Haxxor are all returned as a base-64 string. In some cases when a method is able to support decryption, the key and IV used to encrypt will also be included within the returned hash, such as with AES in the format "key:iv:cipher". This is done so later decryption is possible. Haxxor is more meant as a tool to demo encryption/decryption, and is not meant to be 100% secure. Haxxor is however open-source, so you are able to modify the underlying code to make it more secure. Features Support for multiple methods of encryption and decryption Simple factory class to choose a method Ability to cycle a cipher through multiple methods of decryption when method is unknown Can check if some plain text matches a passed cipher Comes with a console application Comes with a simple real-time encryption/decryption GUI SHA 1, 256, 384, 512 AES 128, 256 MD5 RIPEMD 160 More to come soon. Examples To use the GUI, open up the Haxxor GUI application. Console > Haxxor.Console encrypt aes128 "Hello, world!" > Haxxor.Console decrypt aes128 "hash" > Haxxor.Console list > Haxxor.Console help Code To use the framework, reference the Haxxor.Framework in your project. // Retrieve a module for AES128 encryption var module = HaxxorFactory.GetByType(EncryptionType.AES128); // Retrieve a module by hash // (this is determined by the tag that Haxxor places on a hash, ie "AES128") var module = HaxxorFactory.GetByType(EncryptionType.AES128); var hash = module.Encrypt("Hello, world!"); module = HaxxorFactory.GetByHash(hash); Console.WriteLine(module.Decrypt(hash)); // Cycle a cipher through all possible decryption methods, if the original method is unknown var attempts = HaxxorFactory.Cycle("some hash here"); // Each module also has a property informing if the method supports decryption var module = HaxxorFactory.GetByType(EncryptionType.SHA1); Console.WriteLine("Do I support decryption?" + module.IsDecryptable); //Finally, you can validate some plain text matches a cipher var module = HaxxorFactory.GetByType(EncryptionType.AES128); var hash = module.Encrypt("Hello, world!"); var matches = module.Validate("Hello, world!", hash); Page 3 You can't perform that action at this time.