STEP 1 (BRUTEFORCE)
The Challenge Savoury at first glance takes 15 random passwords from a wordlist of 100 passwords which is provided and then it performs a few simple hashing operations on them after adding a random suffix, a salt and a random digit, these are added at the front if the iteration is odd or to the back if it is even. We need to find the passwords, the suffix and the digits. For this we can simply bruteforce possible combinations of suffixes and digits on the passwords provided from the wordlist after hashing and comparing it to the output provided to find the exact combination of password, suffix and digit. From the description we can assume the salt is `"pepper"` which is made quite obvious and is indeed the salt. Now to bruteforce we can write a simple script like:

```
import hashlib

possible_passwords = [
"travy", "babybaby", "whyyyyy", "Utube", "aliceinwonderland",
"maxtergalindo", "aryamdon", "decjanfeb", "Nebunika",
"calbaby", "lizzywithlucyha", "jakedogg", "panc4ke06",
"acj1354236", "mollybuzz", "briansgrl123", "b1n2c3",
"trippersz", "miera5807", "REYPEREZ", "shnuggles", "ahuvah",
"jezewski", "tummytree347", "ccla2255", "22649981916",
"TEODIOANDREA", "3610458", "072117542", "travis2448", "yamazen",
"Loyalty6", "gukbfr", "69010", "mohawk513",
"artacsp10785", "maude22", "fluffy46", "cavyice26", "dorisdej",
"REVO33", "facatativa", "masuilene", "yaye06",
"hineman", "alidol27", "newtitou", "letmein", "servant4God",
"3127872950", "thezombie", "deia123", "amg305",
"ya07di02ra94", "gazelle102", "hirobaby", "112930632",
"rojapel10", "carlobecerra1995", "polllly102", "ahmedhayat",
"jesuschick2010", "geolcoa", "byntsnq", "ventures", "carla123456",
"504jay", "breane", "emilyyy", "greg51",
"smokita", "maeres88", "3dogsrule", "2006-1993", "adinadan",
"44always", "7SILVIARAU", "password123", "saliha",
"ladyt5", "daresk", "emzie18180", "cailing", "mmahmoodh123",
"mallocalloc", "wipes08", "33plates",
"mitapa", "bryanteamo", "bodyart4u", "7506495i", "jorge011503",
"m32008", "0819627238", "gazeebo", "whooo123",
"mrcvrazed45", "paulinarocks", "m01p09p05", "CARMENIG"
]

digits = [str(i) for i in range(10)]
suffixes = [chr(i) for i in range(ord('a'), ord('z') + 1)] \
         + [chr(i) for i in range(ord('A'), ord('Z') + 1)] \
         + ['!', '@', '#', '%', '&']

base_salt = "pepper"

hash_input = input("enter hash: ")
opt = input("Enter 1 for even and 0 for odd: ").strip()

found = "NONE"

for pw in possible_passwords:
    for d in digits:
        for s in suffixes:
            if opt == '1':
                salt = base_salt + d + s
                sha256_hash = hashlib.sha256(hashlib.md5((pw +
salt).encode()).digest()).hexdigest()
            elif opt == '0':
                salt = base_salt + d + s
                sha256_hash = hashlib.sha256(hashlib.md5((salt +
pw).encode()).digest()).hexdigest()
            else:
                continue

            if sha256_hash == hash_input:
                if opt == '1':
                    found = pw + salt
                elif opt == '0':
                    found = salt + pw
                break
        if found != "NONE":
            break
    if found != "NONE":
        break

if found != "NONE":
    print("Match found:", found)
else:
    print("No match found for hash:", hash_input)i
```

You can either crack the hashes one by one or run all 15 together, we get:

```
Index: 0, Password: 'shnuggles', Salt: 'pepper7R', Hash:
7975a9d76aeaace326887eb820e536fab71a7c2436a218c6cf64bee1e540bda1
password: shnugglespepper7R
Index: 1, Password: 'CARMENIG', Salt: 'pepper8y', Hash:
515ee65a543717cba12f2de8efea7e57221a555366a31f9fb1d1173a0b2f498d
password: pepper8yCARMENIG
Index: 2, Password: 'ya07di02ra94', Salt: 'pepper2t', Hash:
6e220d66b819185d4087f61bad01da59d1020101441caad54b78d9e4397fb705
password: ya07di02ra94pepper2t
Index: 3, Password: 'maxtergalindo', Salt: 'pepper8T', Hash:
3888346f9cd340fef37b284b68afec838da412248d942d0529222857bc63c76f
password: pepper8Tmaxtergalindo
Index: 4, Password: 'password123', Salt: 'pepper0e', Hash:
711a5e6a996b9e822d6a9c34bc1781309671b492826c5f3eb301a52cf9bf2faa
password: password123pepper0e
Index: 5, Password: 'hirobaby', Salt: 'pepper0L', Hash:
441a06a97fa7c055d0a6bacecdc5cdde6f7bee4c0a5395171a8cc1649fd39420
password: pepper0Lhirobaby
Index: 6, Password: 'letmein', Salt: 'pepper2#', Hash:
b035debeb585c95677df821f16a65daaa1b8e9c80893ecbe16281db8b23e22d8
password: letmeinpepper2#
Index: 7, Password: 'daresk', Salt: 'pepper2X', Hash:
97add6cd5c1a3796a74fa922e6220329cf8184ab1bf9e9201d12361513a1d748
password: pepper2Xdaresk
Index: 8, Password: 'mollybuzz', Salt: 'pepper4X', Hash:
d252448dac9a756a9af748a84c6e419d44a1a966ca3eb6416d844bce48abe790
password: mollybuzzpepper4X
Index: 9, Password: 'panc4ke06', Salt: 'pepper4R', Hash:
4e17f090b0c9a914c4c31ea41c87dffa75b8ec5d0bee1cee3896c94aae79c705
password: pepper4Rpanc4ke06
Index: 10, Password: 'briansgrl123', Salt: 'pepper7w', Hash:
0593282c56c2754c13a69d18de8f2d034a97d97bfddfad039e1f73ae7cae98e8
password: briansgrl123pepper7w
Index: 11, Password: 'thezombie', Salt: 'pepper1p', Hash:
ac4ebe9ae15a0e3b26a1e06265649a208e6b4d23ac8f3475f9e6d77d4086db67
password: pepper1pthezombie
Index: 12, Password: 'gazelle102', Salt: 'pepper4p', Hash:
8f913c56515757fc1ab850c796ee85b58b29488dc16a3bb6bb22b864f473f68a
password: gazelle102pepper4p
Index: 13, Password: 'carlobecerra1995', Salt: 'pepper8@', Hash:
01f061c145a396f17a835b55ebfca154cc945a706c079153982ccf54ccbea4c9
password: pepper8@carlobecerra1995
```

```
Index: 14, Password: 'emilyyy', Salt: 'pepper3Z', Hash:
35bce8118f76ee174f5ad68fd98ab57efc9fe1eaf2a1f4a2cd43f27fe31c9750
password: emilyyypepper3Z
```

STEP 2 (GENERATING THE FLAG)

from the results we can generate the flag with the help of the instructions provided in the description. The flag obtained is "DJSISACA{s7R_C8y_y2t_m8T_p0e_h0L_l2#_d2X_m4X_p4R_b7w_t1p_g4p_c8@_e3Z_pepper}".