DOMANDE PROGRAMMAZIONE ESAME ORALE RICCIO APPELLO GENNAIO 2021 PRIME 18 PERSONE:

Persona 1:

- Che cos'è la ricorsione? **una funzione che nella sua definizione chiama se stessa**
- Differenza tra ricorsione e iterazione? **una funzione iterativa dà parte della soluzione al problema ad ogni passo fino a raggiungere la soluzione totale. una funzione ricorsiva cerca di ridurre il problema gradualmente fino ad arrivare al caso base e dare una soluzione totale dopo un effetto domino inverso creato dalle sue precedenti chiamate**
- Quando si usa la ricorsione? Pur sapendo che è meno efficiente? **la ricorsione permette di scrivere un codice più breve e intuibile. il problema è che la ricorsione pesa sul record di attivazione perchè ogni volta che la funzione viene chiamata viene allocata in memoria.**
- C'è un modo per mantenere il meccanismo ma evitando di attivare un record di attivazione? **sì, creando uno stack esplicito che permette di simulare la chiamata di una funzione**
- Che cos'è uno stack? **è una struttura dati astratta che ha modalità di accesso di tipo LIFO**
- Ci sono casi in cui una funzione ricorsiva non si può fare in modo iterativa? **si dovrebbe poter sempre fare ma determinate volte diventa scomodo usare le funzioni in modo iterativo perchè il codice diventa illegibile**
- Definizione di complessità di tempo? E del problema? **la complessità temporale di un algoritmo sarebbe il numero di operazioni elementari eseguite dall'algoritmo. le operazioni elementari hanno un tempo di esecuzione fisso.**

Persona 2:

- Cosa sono i puntatori e come funzionano? **è una variabile che contiene un indirizzo di una locazione di memoria. quando noi dichiariamo una variabile a quella variabile verrà associato un indirizzo di memoria, il puntatore contiene appunto l'indirizzo di memoria di quella variabile**
- Allocazione di memoria dinamica invece che statica? **si alloca memoria dinamicamente quando non sappiamo quanto spazio è necessario in memoria. per allocare spazio dinamicamente si possono usare due tipi di funzione. la malloc e la calloc. la calloc inizializza gli elementi allocati a 0, mentre la malloc no.**
- Cos è la ricerca binaria e come funziona? **si usa per cercare un elemento in un arrey già ordinato. si verifica che l'elemento che si sta cercando non sia l'elemento mediano dell'arrey. se è più grande, si cerca nuovamente l'elemento tra med+1 e n, se è più piccolo tra 0 e med-1.**
  - Complessità? **log2 di n**
- Conviene prima ordinarlo e poi fare la ricerca binaria? **non si può fare la ricerca binaria se l'arrey non è ordinato**
- Complessità counting sort? **O(n+k)**
  - Qual è il principale svantaggio? **dobbiamo usare due arrey di supporto, quindi pesa in memoria**

- Si può avere un counting sort efficiente senza allocare tutta questa memoria? **idk**

Persona 3:

- Che cos'è una struttura ricorsiva? **una struttura ricorsiva che al suo interno contiene un riferimento a se stessa**
  - Quando definiamo una struttura ricorsiva, perché inseriamo un puntatore a next invece di un solo next? **perchè deve essere un puntatore a struttura non una variabile del tipo della struttura**
- Quanta memoria occupa un record? **si deve fare la somma dei tipi contenuti nella struttura**
- Come fa a capire una sizeof di una struct? **somma dei tipi**
- Quanto è grande un puntatore? **vale quanto un int, quindi 4 byte su 32 bit e 8 byte su 64bit**
- Che cos'è una coda? **una struttura astratta con accesso di tipo FIFO, cioè: noi possiamo fare due operazioni su una cosa estrarre e aggiungere, essendo di tipo FIFO quando estraiamo noi prendiamo il più vecchio elemento aggiunto**
- Che cos'è una lista circolare? **è una lista che ha come puntatore a next dell'ultimo nodo l'indirizzo del primo elemento della lista**
  - A cosa bisogna stare attenti in una lista circolare? **a qual è l'ultimo nodo. in una lista normale il puntatore all'ultimo nodo punta a NULL, nella lista circolare punta all' elemento del primo. quindi si mette un nodo finto alla testa (sentinella)**
- Quanto costa inserimento e cancellazione in una coda? **costa 1, perchè essendo una lista doppiamente concatenata, l'elemento in testa ha un puntatore alla coda, quindi non devo prima scorrere tutta la lista per trovare la coda, mi basta un unico spostamento**

Persona 4:

- Perché inserire e togliere una coda costa o(1)? **in una lista doppiamente concatenata, l'elemento in testa ha un puntatore alla coda, quindi non devo prima scorrere tutta la lista per trovare la coda, mi basta un unico spostamento**
- Cos'è una sentinella in una lista? **è un nodo speciale che non contiene informazioni ma serve solo a marcare l'inizio o la fine della lista**
  - Quando si usa la sentinella? **in una lista circolare, perchè dovrebbe andare a semplificare alcune operazioni e serve a controllare che i vari nodi siano collegati bene**
- In una lista dl, cioè bidirezionale, come si fa la cancellazione del nodo? **trovi il nodo, fai collegare il nodo.next precedente al nodo da eliminare al nodo.prev successivo del nodo eliminato e il puntatore nodo.prev del successivo al nodo.next del precedente**
  - Quali sono i casi? **i casi sono: lista vuota, lista con un solo elemento w una lista con n elementi**
  - Una coda può essere eliminata dal centro?
- Cos'è un array di array? **una matrice**
- Data una matrice M di 4x3, M che tipo è? **int ** cioè un puntatore a puntatori**
- Distinzione tra puntatore a vettore di 3 interi e puntatore a puntatore di interi? **int *p=&vettore[0]; e int **v;**

Persona 5:

- Che cos'è lo scope di una variabile? **la parte di codice in cui quella variabile è visibile**
- Che cos'è una variabile statica? **una variabile statica è locale ad una particolare funzione. è inizializzata una sola volta, la prima volta che tale funzione viene chiamata e il suo valore resta inalterato quando si esce dalla funzione, per cui quando si richiama nuovamente la funzione tale variabile ha ancora il valore assegnatogli precedentemente. per definire statica una variabile è sufficiente anteporre la parola static alla dichiarazione della variabile.**
  - Qual'è il vincolo di una variabile statica? **va inizializzata**
- Che cos'è una constante? **una variabile che ha un valore fisso che non può essere modificato**
  - Perche si usa? **per evitare che il dato venga modificato durante il programma**
- Differenza tra dichiarare una define e definire un identificatore const? **della define se ne occupa il processore mentre della const se ne occupa il compilatore**
- Principio del minimo privilegio? **è un concetto di sicurezza delle informazioni che permette ad un utente un accesso minimo per accedere alle parti del codice che gli serve**
- Ci sono casi in cui è meglio usare un ciclo for di un while? **il for va usato quando oltre ad avere una condizione da verificare ho una che un'operazione di passaggio da svolgere ogni volta che il ciclo riparte, mentre nel while c'è solo la condizione di veridicità**
- In che modo scrivo un ciclo infinito con un while? **per realizzare un ciclo infinito basta far sì che la condizione di ripetizione del ciclo sia sempre vera**
  - Per uscire da questo loop? **con l'istruzione di break**
- Cosa fa il continue? **interrompe l'iterazione di un ciclo per passare a quella successiva**
- Complessità caso migliore del bubble sort? **n-1**

Persona 6:

- Come funziona la definizione di un nuovo tipo? **typedef**
  - Come usare typedef come tipo e come dichiarazione?(DA VEDERE LA DOMANDA)
- Che cos'è un puntatore a void? **è un puntatore di cui non si conosce il tipo del dato a cui punta**
  - Se non avessi il void in una malloc cosa succede? **_____?**
- Come si chiama l'operazione che trasforma un puntatore a void in un tipo diverso? **casting**
- Differenza tra cast implicito ed esplicito? **il casting implicito viene eseguito automaticamente dal compilatore, quello esplicito è eseguito forzatamente dal programmatore ex: malloc**

- Quanti valori può comprendere una variabile di tipo char? **[−128,127] (256 valori)**
- Con un unsigned char? **0 a 255**
- Differenza tra char e unsigned char? **l'intervallo di rappresentazione**
- Che cos'è il complemento a 2? **un tipo di rappresentazione numerica**

Persona 7:

- Come funziona il counting sort? **il cs è un algoritmo di ordinamento not in place. serve nel momento in cui voglio ordinare un vettore che ha più ripetizioni. l'ordine iniziale degli elementi degli arrey è indifferente in quanto non andiamo a operare sull'arrey iniziale ma con due arrey di supporto. (per forza su un foglio)**
- Complessità? **se consideriamo k la dim del vettore di supporto e n quello da ordinare diciamo che ha complessità O(k+n)**
- Modo più efficiente per ordinare una lista? **bubble sort (costa meno)**
- Che cos'è la sequenza di Fibonacci? **è una successione di numeri interi di cui i primi due numeri sono 1 e 1 e i seguenti sono la somma dei precedenti. si può rappresentare come una funzione ricorsiva con 2^n chiamate in quanto si scrive:**
  long fib(long n){
  if(n<=1)
  return 1;
  else
  return fib(n-1)+fib(n+2);
  } quindi la funzione viene chiamata ogni ciclo due volte, è molto pesante
  - Come migliorarlo? **si usa uno stack esplicito che simula le chiamate della funzione**
- Che cos'è la Torre di Hanoi? **è un rompicapo matematico in cui abbiamo tre paletti e dei cerchi impilati su uno di questi in ordine decrescente, lo scopo del rompicapo e far passare i cerchi per tutti e tre i paletti facendo in modo che siano sempre in ordine decrescente.**
  - Complessità? 2^n-1, oppure o(2^n) **la seconda**
- Che cos'è una union? **è un tipo di struttura in cui ci possono essere membri di diverso tipo ma possono essere utilizzati solo uno per volta**

Persona 8:

- Allocazione dinamica della memoria? **quando non sappiamo ancora la memoria precisa che un dato occuperà. si usa la funzione malloc int *t=(int*)malloc(sizeof(int)); si fa un casting esplicito altrimenti il risultato sarà un puntatore a void**
- Puntatore a void?
- Posso deallocare un vettore dichiarato cosi: int v[10]? **nop, dovrebbe essere un puntatore costante**
  - Chi si occupa di deallocarlo? **il sistema operativo**
- Aritmetica dei puntatori sui vettori? **il nome di un vettore non è altro che un puntatore al primo elemento (quello in posizione 0)**

- Creo un array dinamico con la malloc, poi tramite aritmetica dei puntatori faccio *p+5, e dealloco con una free *p+5, qual è il problema? **se fai free(*p+5) non ha senso perché la free ha bisogno di un puntatore e non di un *p (che sarebbe l'elemento puntato), dovrei fare per esempio q=p+5 e poi fare free(q)**
- Differenza tra una stringa e array di caratteri? **la stringa è un arrey di caratteri con il carattere terminatore**
- Posso contare quante volte una funzione ricorsiva viene chiamata? **si può mettere un contatore che viene aumentato ogni volta che la chiamata viene ripetuta**
- Differenza tra static ed exstern? **dichiarare una var come static serve a fare in modo che non venga eliminata dopo che è stata utilizzata in una funzione, extern serve ad estendere la visibilità di una var esterna**
- Differenza tra coda e lista? **la coda è una struttura fifo, cioè che il primo elemento ad essere aggiunto sarà anche il primo ad uscire mentre la lista è una struttura ricorsiva**

Persona 9:

- Che cos'è la torre di Hanoi e complessita? **è un rompicapo matematico in cui abbiamo tre paletti e dei cerchi impilati su uno di questi in ordine decrescente, lo scopo del rompicapo e far passare i cerchi per tutti e tre i paletti facendo in modo che siano sempre in ordine decrescente. ha complessità O(2^n) dove n sono il numero dei dischi**
- Complessità problema di fibonacci? **sempre 2^n**
  - Cicli in fibonacci? **----**
- Che cos'è la complessità? **misura del numero di passi che devo eseguire per risolvere un problema**
- Che cos'è la complessità del problema? **la complessità del migliore algoritmo che risolve quel problema**
- Cos'è una stack? **è una struttura dati astratta di tipo LIFO, last in first out. la si può immaginare come una pila di libri**
- È possibile implementare uno stack in uno array? **sì se gli si da una dimensione finita e quindi un indice massimo di elementi che si possono aggiungere**
- Come si calcola la dimensione di un tipo? **con sizeof**
  - Con un record come funziona? **è la somma delle dimensioni dei tipi contenuti nel record**
- Ciclo infinito con il for? **un for senza nulla scritto nelle tonde** `for(;;)`
  - Per uscire dal ciclo infinito? **si usa l'istruzione di break**
- Come funziona il bubble sort? **il bubble sort è un algoritmo di ordinamento in place in cui si ordina scambiando le coppie successive fino a che non si ha l'ultimo el dell'arrey che è il minore o maggiore a seconda del tipo di ordinamento. poi si ripetono gli scambi da capo**
  - Caso peggiore? **avere un arrey ordinato al contrario di come lo vogliamo noi**

Persona 10:

- Come funziona la ricerca binaria? **si può fare solo con un arrey ordinato già. confrontiamo l'elemento cercato con quello mediano dell'arrey e**

vediamo se ricercarlo poi nell'intervallo precedente a quello mediano o successivo.
    - Probabilità che ho nel beccare il numero nell'array al primo colpo nella mediana? **1/n**
- Se ho un vettore di caratteri posso fare la ricerca binaria? **sì perchè nel codice ascii i caratteri hanno un ordine**
- Posso usare i caratteri come tipo intero? **unsigned char**
- Che cos' è un tipo ricorsivo?
- Si può dichiarare una variabile di tipo void?
    - Perche?
- Come si effettua una cancellazione in una S_L?

Persona 11:

- Passaggio di parametri del main/funzione? **la funzione main viene chiamata dal sistema operativo non appena il prog viene avviato, è possibile passare i parametri dalla linea di comando utilizzando il main con questi argomenti: int main(int argc, char *argv[]) di cui il primo è il numero di argomenti inseriti, mentre il secondo è un vettore di stringhe di cui ogni parametro è una stringa.**
- Se passo in parametro un array? **lo copiamo con strcpy**
- Quando dichiaro const si riferisce a diversi elementi? **----------**
- Può succedere che si può cambiare il contenuto e non il valore? **----------**
- Cosa significa deferenziare un puntatore? **con * che è l'operatore dideferenziazione. dereferenziare un puntatore significa ottenere il valore che è memorizzato nella posizione di memoria puntata dal puntatore**
- Perchè non si deferenzia un puntatore a void? **esso può puntare qualsiasi variabile, indipendentemente dal tipo, ma non può modificare la variabile stessa senza un'intenzionale operazione di cast**
- Cosa significa usare uno stack esplicito per la ricorsione? **simulare le chiamate di una funzione nel codice stesso**
- Svantaggio della ricorsione rispetto alla iterazione? **la ricorsione pesa di più in memoria perchè ogni volta che una funzione viene chiamata viene memorizzata nel record di attivazione**
- Si possono contare gli elementi in una coda? **----------**
    - Come posso evitare però di eliminare la coda? **----------**

Persona 12:

- Come avvieni il confronto tra 2 stringhe? **di uguaglianza, confronto i caratteri delle stringhe per vedere se sono sia uguali che nella stessa pos (anche il terminatore nullo)**
  si può fare anche il confronto di ordine

    - Che funzione si occupa del confronto? **strcmp, confronta la pos dei caratteri con quelli della tabella ASCII, le lett min<maius**
- Ho 2 stringhe a e b, e voglio trovare tutti i caratteri in a presenti in b (le stringhe sono ordinate) come faccio? **ricerca di una sottostringa con strstr**
    - Si può fare di meglio?
- Passaggio per parametri da linea di comando? **int main (int argc, char *argv[])**
- Che cos'è un'espressione in c? **i costrutti che compongono un programma, una combinazione di operatori e operandi**
    - Qual è l'espressione più semplice? **i++**
- Rappresentare valori in c? vero e falso?
- Che cos'è null? **è un valore che indica che non ci sono valori**
- Come avviene la cancellazione dell' ultimo nodo in una D_L? **----------**

Persona 13:

- Allocazione dinamica? **serve a conservare un tot di memoria quando non sappiamo ancora quanta ce ne servirà esattamente**
    - Perchè la facciamo?
    - Cosa non sappiamo?
- Come si scrive una malloc? **se per esempio vogliamo allocare memoria per una var puntatore**

  int *p;

  p= (int*)mallloc(sizeof(int))
    - Perchè si fa un cast? **perchè devo specificare il tipo di puntatore che voglio in uscita, altrimenti mi viene restituito un void* senza tipo**

Persona 14:

- Differenza tra malloc e calloc? **la calloc alloca un numero specifico di byte e li inizializza a 0, mentre malloc alloca solo dello spazio senza inizializzarlo**
- Se dichiaro un vettore staticamente, che succede se non inizializzo tutte le celle? **le celle non inizializzate vengono inizializzate automaticamente a 0**
- Modo comodo per inizializzare un vettore senza il ciclo for? **inizializzare l'arrey a 0, int a[5]={0}, oppure int a[5]={1,2,3,4,5} io inzio a inserire da a[0]**
- Aree di memoria dove vanno le variabili statiche e dinamiche? **stack e heap**
- Cosa ci finisce nello stack di sistema? **var locali del main**
- Come viene gestita una chiamata a funzione? **viene creato un record di attivazione che è gestito come uno stack, ogni volta che la stessa funzione viene chiamata si aggiunge al record di attivazione. viene creato un record di attivazione diverso per funzioni diverse**
    - Come e dove riserva la memoria? **nello stack(?)**
- Come funziona il passaggio di parametri? **per valore, nel caso di funzioni, e per riferimento nei puntatori anche se è un riferimento fake perchè in c non esiste**
- La ricorsione ha comunque a che vedere con lo stack di attivazione, perche? **per la chiamata dell funzioni**

- Cosa succede quando una chiamata ricorsiva termina? **il record di attivazione viene svuotato e deallocato**
- Principio della ricorsione su cosa si basa? **sul risolvere un problema dividendolo in problemi sempre più piccoli per arrivare ad un caso base che sappiamo risolvere**
    - Come funziona?

- Che cos'è la serie di fibonacci? **è una serie numerica di numeri interi dove i primi due numeri sono 1 e 1 e i successivi sono la somma dei precedenti**
    - La sua funzione ricorsiva?
      long fibo(long n){
      if(n<=1)
      return 1;
      else
      return fibo(n-1)+fibo(n+2);

    - Quando si ferma la ricorsione? **quando si arriva al caso base**
    - Complessità fibonacci? **O(2^n)**
        - Perchè? **perchè la funzione viene chiamata due volte ogni volta**

Persona 15:

- Ricerca binaria cos'è?
    - Quando si usa? **quando dobbiamo cercare un el in un vettore già ordinato**
    - Complessità? **O(log2 di n)**
    - È possibile implementarla in modo ricorsivo?
        - Complessità della versione ricorsiva? **la stessa di quella iterativa perchè la chiamata viene eseguita sul vettore dimezzato**
- Che cos'è una coda? **una struttura dati astratti che ha modalità FIFO**
    - Costo eliminazione e inserimento? **logn**

Persona 16:

- Strutture dati ricorsive? **liste**
- Ordinamento per ordinare una lista? **bubble sort**
- È possibile implementare una lista con un array? **sì, ma sarebbe scomodo**
- Cosa significa complessità O(n)? **che dobbiamo fare n passi per risolvere un problema**
- Che cos'è O-grande? **il comportamento asintotico di una funzione**

Persona 17:

- Counting sort?
    - Svantaggio? **dobbiamo usare due arrey di supporto quindi pesa sulla memoria perchè è un not in place**
- Dichiarare un nuovo tipo in C? **typedef**

    - Dove viene soprattutto utilizzato? **nelle struct**
- Passaggio di parametri? **per valore, nel caso di funzioni, e per riferimento nei puntatori anche se è un riferimento fake perchè in c non esiste**
- Cosa succede quando passiamo il puntatore? **la funzione può utilizzare il puntatore per modificare la variabili**
- Parametri attuali e formali? **i parametri formali sono quelli tra parentesi nella dichiarazione della funzione, quelli attuali o argomenti, sono quelli tra parentesi nella chiamata**
- Cosa succede se passo un array alla funzione? **stiamo passando il puntatore al primo elemento di quell'arrey**
    - Se non voglio modificare il puntatore? ??

Persona 18

- Cos'è una stack?**è una struttura dati astratta con modalità LIFO, la si può vedere come una pila di libri**
- Che cos'è una union? **UNA STRUTTURA I CUI MEMBRI SI POSSONO UTILIZZARE SOLO UNO ALLA VOLTA**
- Cosa significa implementare Una ricorsione con lo stack esplicito? **simulare le chiamate della funzione nel codice stesso**
- Se ho una stringa di caratteri e voglio girarla al contratrio, come faccio? **mettiamo due contatori, uno di decremento e uno di incremento e scambiamo, primo e ultimo menìmbro, e così via**

Qs. 1 – (5 + 3 punti): si risponda alle seguenti domande riportando su un foglio bianco il numero della domanda e al lato di ciascuna numero una V se si considera la risposta vera o una F se si ritiene che la risposta sia falsa.

1) Il compilatore ed il linker traducono il codice sorgente in un eseguibile
2) Il bit più significativo è detto *LSB*
3) Nel sistema binario il valore di una cifra dipende dalla sua posizione
4) Con n bit si rappresentano 2^n-1 valori
5) La direttiva *#define* definisce costanti simboliche
6) *sizeof*(char) restituisce 1 su qualsiasi PC
7) La direttiva #define definisce costanti simboliche
8) L'operatore -= sottrae l'r-value al l-value
9) La funzione strlen calcola la lunghezza di una stringa
10) In C una stringa è una matrice di caratteri
11) Un numero binario di 4n bit corrisponde ad un numero esadecimale di n cifre
12) Il complemento a 1 di un numero A è B = 2^n-1+A
13) Nel complemento a 2 lo zero ha una sola rappresentazione
14) In un ciclo for l'istruzione continue interrompe il ciclo
15) Il token *piano_forte* non è un identificatore valido
16) Il campo di visibilità di una variabile è detto *scope* della variabile
17) L'operazione di casting forza una conversione esplicita di tipo
18) Un costrutto if/else si può sempre riscrivere con un costrutto switch/case/default
19) La tabella ASCII comprende 256 simboli
20) Per copiare due record, si deve copiare ciascun membro

Sia dato il vettore V={40, 33, 1, 21, 4, 50}. Si mostri lo stato del vettore ad ogni passo dell'algoritmo *Selection Sort*.

Iterazione	V[0]	V[1]	V[2]	V[3]	V[4]	V[5]
1	40	33	1	21	4	50
2						
3						

Qs. 2 – (5 punti): si indichi l'output del seguente programma, così come apparirebbe sul monitor di un computer, ovvero tenendo conto anche della formattazione.

```c
#include <stdio.h>
#include <string.h>

char str[] = "Programmazione I";

int main()
{
	int i,n;

	n = strlen(str);

	for(i=0; i<n; i++){
		if(i%2)
			continue;
		printf("%c ", str[i]);
	}

	return 0;
}
```

Qs. 3 – (5 punti): si individuino gli errori nel seguente programma. Per le righe, in cui è presente un errore, si scriva sulla linea corrispondente l'errore individuato. Per le righe, in cui non sono presenti errori, si scriva a fianco "Corretto".

```
1   #include >stdio.h<
2   int main(double argc, char argv) {
3       struct {double a; float b;}p;
4       p.a = &p.b;
5       for (i = 0; i < 5; i++) {
6             if (i % p.a == 0)
7               printf("%f ", p.a);
8       )}
9       return p.a;
10  }
```

Qs. 4 – (12 punti):

Scrivere un programma C che:
1) Prende in input dalla linea di comando una parola
2) Definisce un nodo *Carattere* come un record che contiene un carattere e un puntatore al nodo successivo
3) Implementa la funzione *int* ***crea_lista**(char parola[])* che prende in input una parola e restituisce il puntatore alla testa di una lista di caratteri

Input e Output
Input: imprescindibile La lista è: i->m->p->r->e->s->c->i->n->d->i->b->i->l->e->NULL