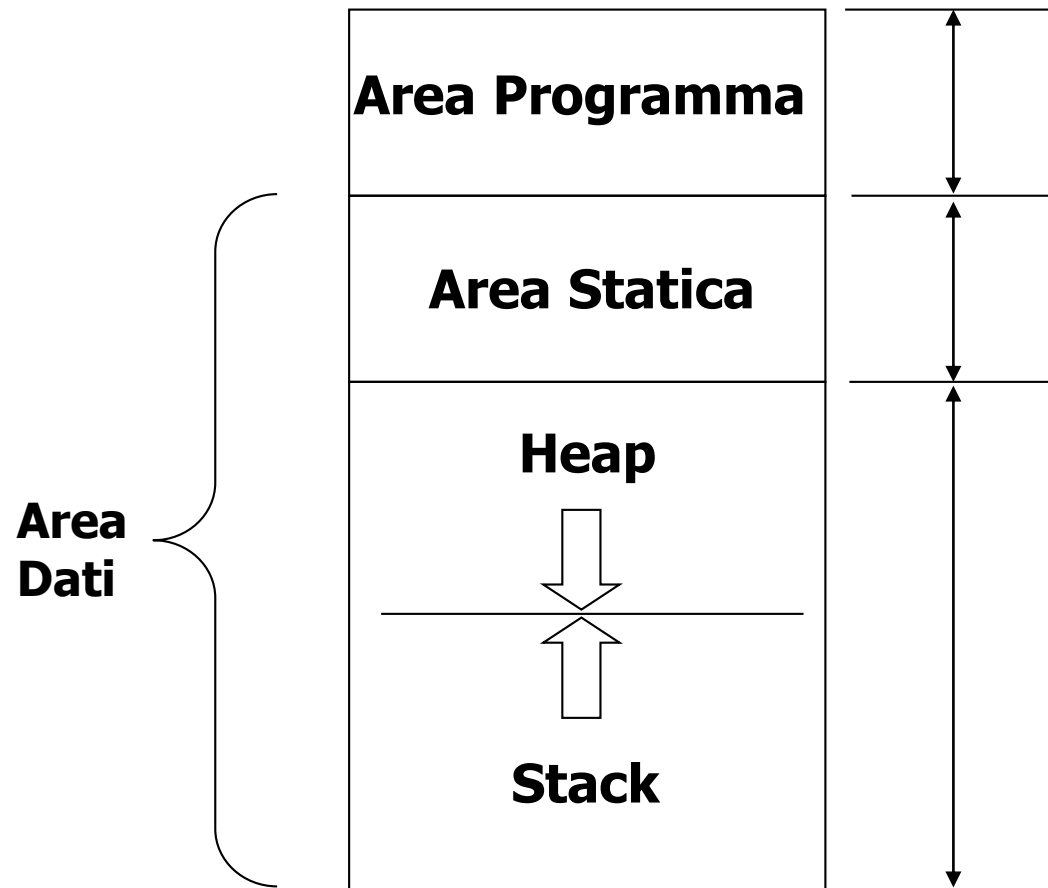




Record di Attivazione e Parametri Formali di una Funzione

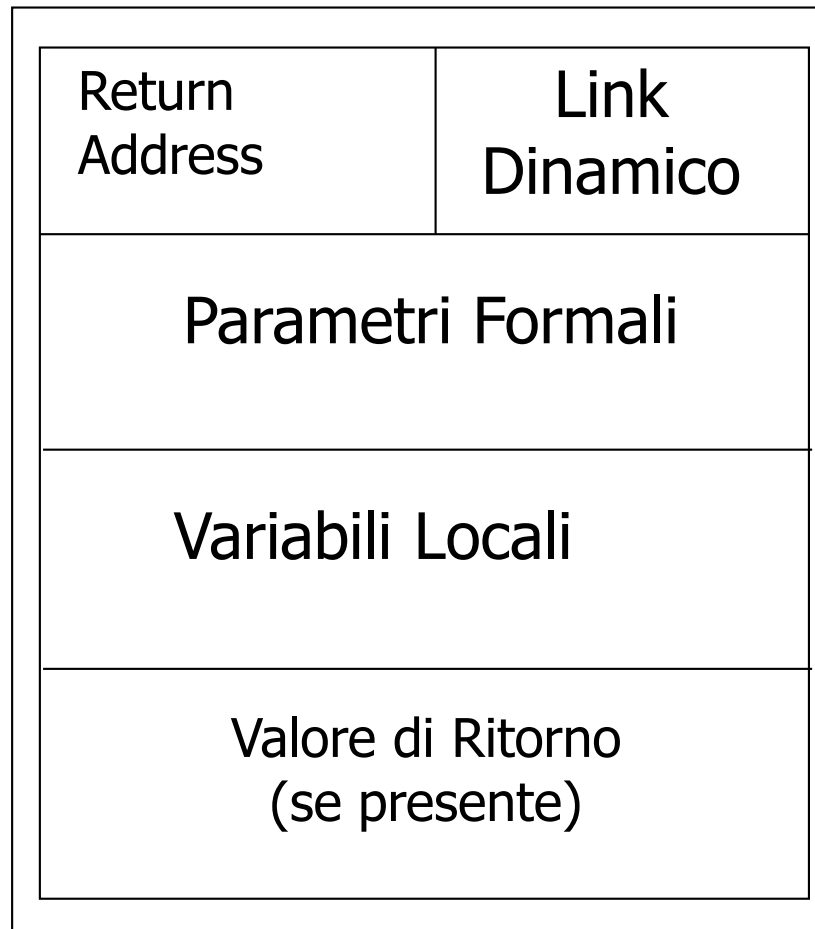
Prof.Ing.S.Cavalieri

Area di Memoria Stack





Record di Attivazione



Record di Attivazione

Record di Attivazione: Link Dinamico

```
#include <stdio.h>
int x;
```

```
void R (int a) {
    x=a;
}
```

```
void Q (int b) {
    R(b);
    printf("\n%d",b);
}
```

```
void P (void) {
    int a=10;
    Q(a);
    printf("\n%d",a);
}
```

```
int main(void){
    P();
}
```

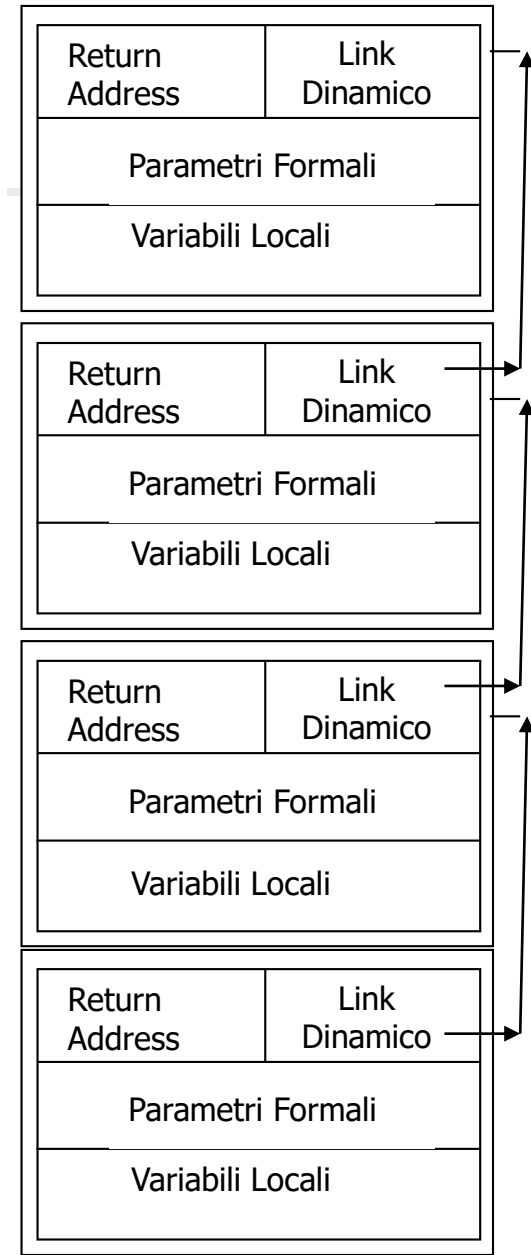
Stack

main()

P()

Q()

R()



Record di Attivazione: Return Address

```
#include <stdio.h>
int x;

int R (int a) {
    return a;
}

void Q (int b) {
    int a;
    B a=R(b);
    if (!a) printf("\nCiao");
}

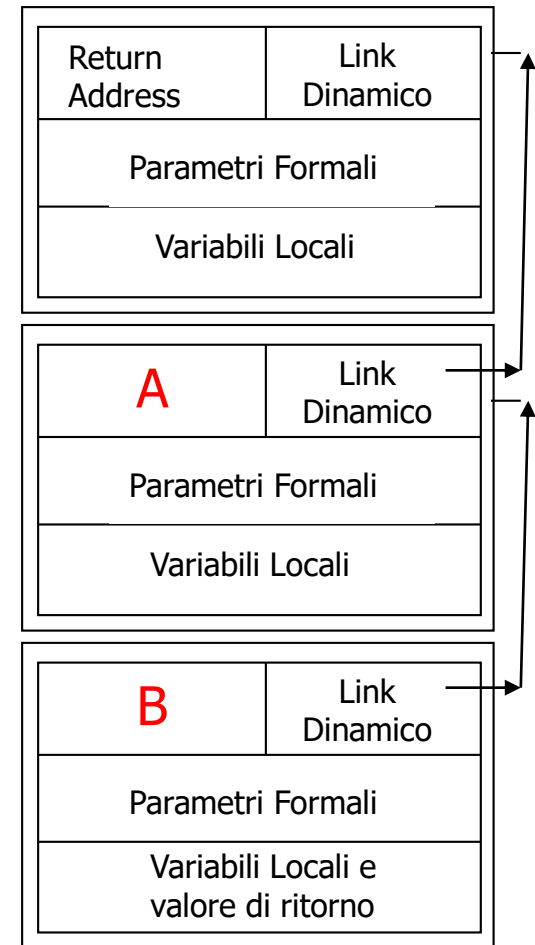
int main(void){
    Q(5);
    A printf("\nFine ");
}
```

Stack

main()

Q()

R()



Record di Attivazione: Parametri e Valore Ritorno

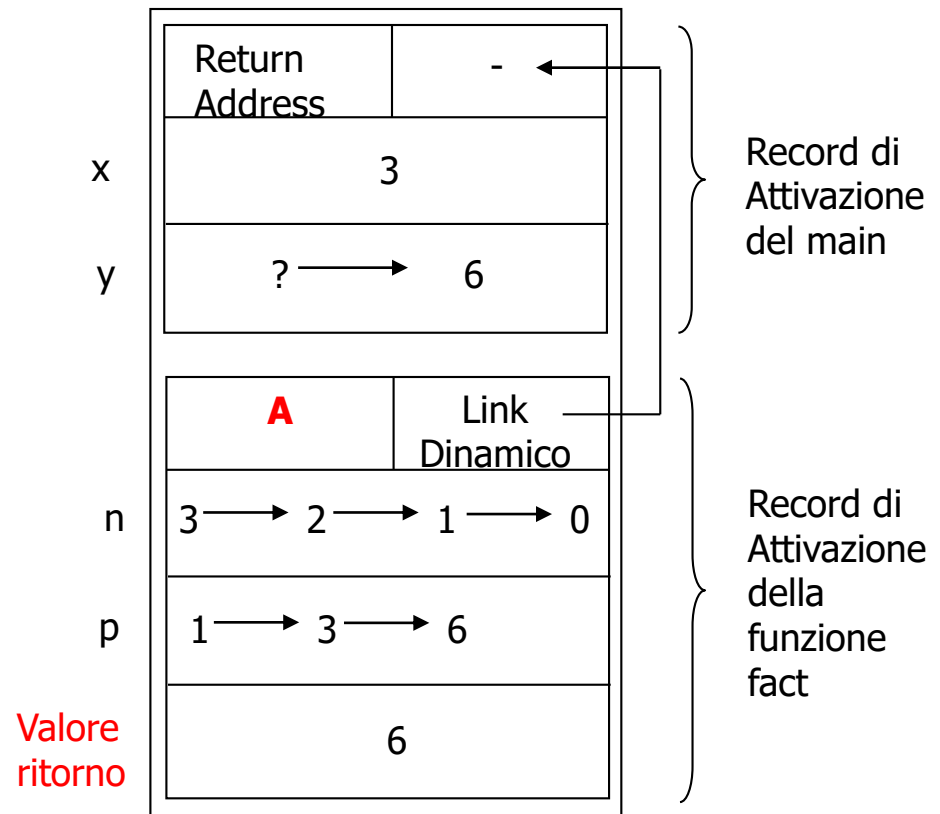
```

#include <stdio.h>

double fact (unsigned long);

int main(void){
    int x=3, y;
    A y=fact(x);
}

double fact (unsigned long n) {
    double p=1;
    while (n>0) p*=n--;
    return p;
}
    
```

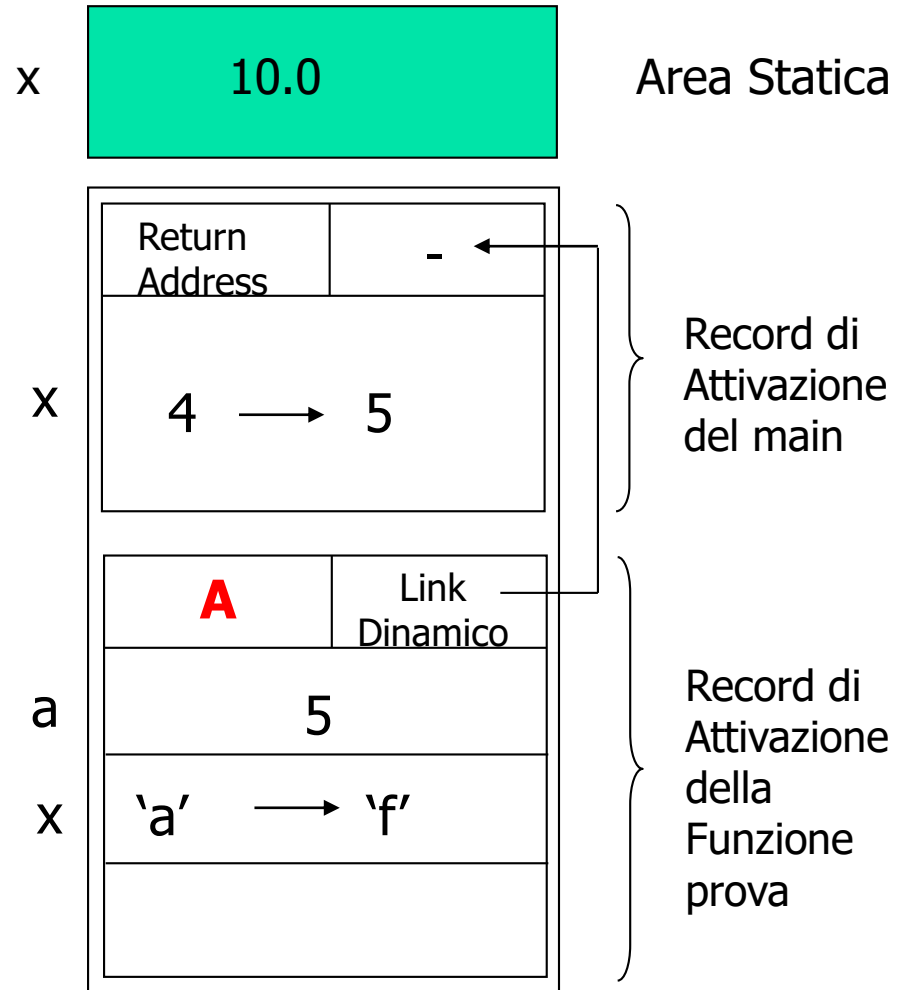


Record di Attivazione: Variabili Locali

```
#include <stdio.h>
float x=10.0;
void prova ( int );

int main(void){
    int x=4;
    x++;
    prova(x);
    A printf("%d \n", x);
}

void prova (int a){
    char x='a';
    x+=a;
}
```



Passaggio Parametri : Passaggio per Valore !

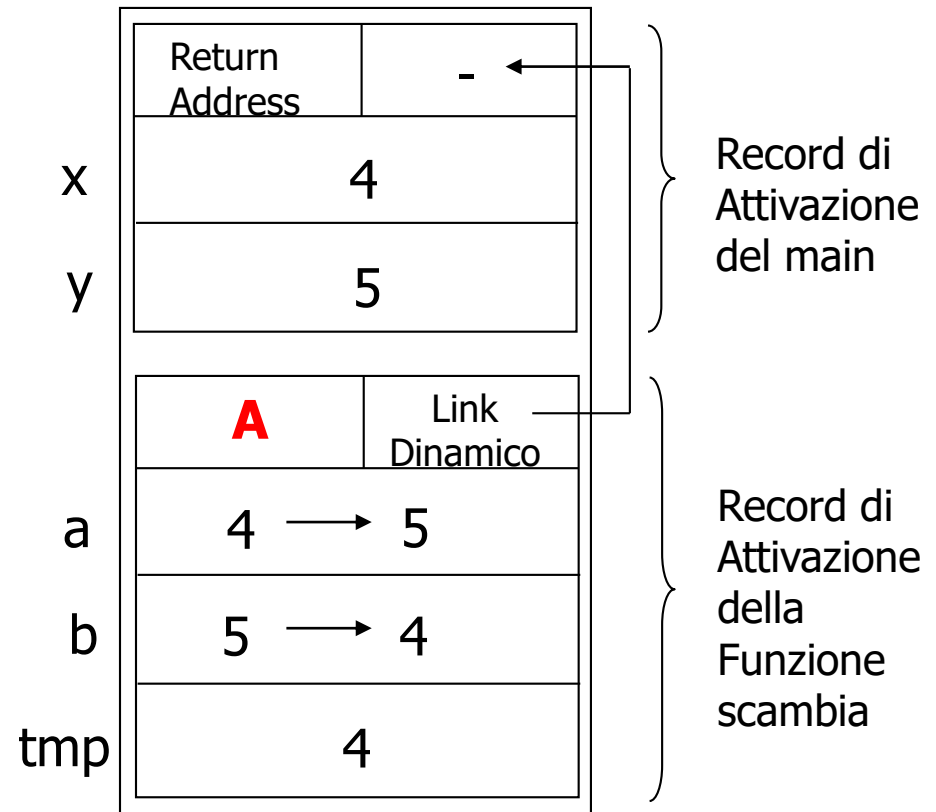
```
#include <stdio.h>

void scambia ( int , int );

int main(void){
    int x=4, y=5;

    scambia(x,y);
    A printf("%d %d \n", x, y);
}

void scambia ( int a, int b) {
    int tmp=a;
    a=b;
    b=tmp;
}
```



Passaggio Parametri: Soluzione passaggio per &

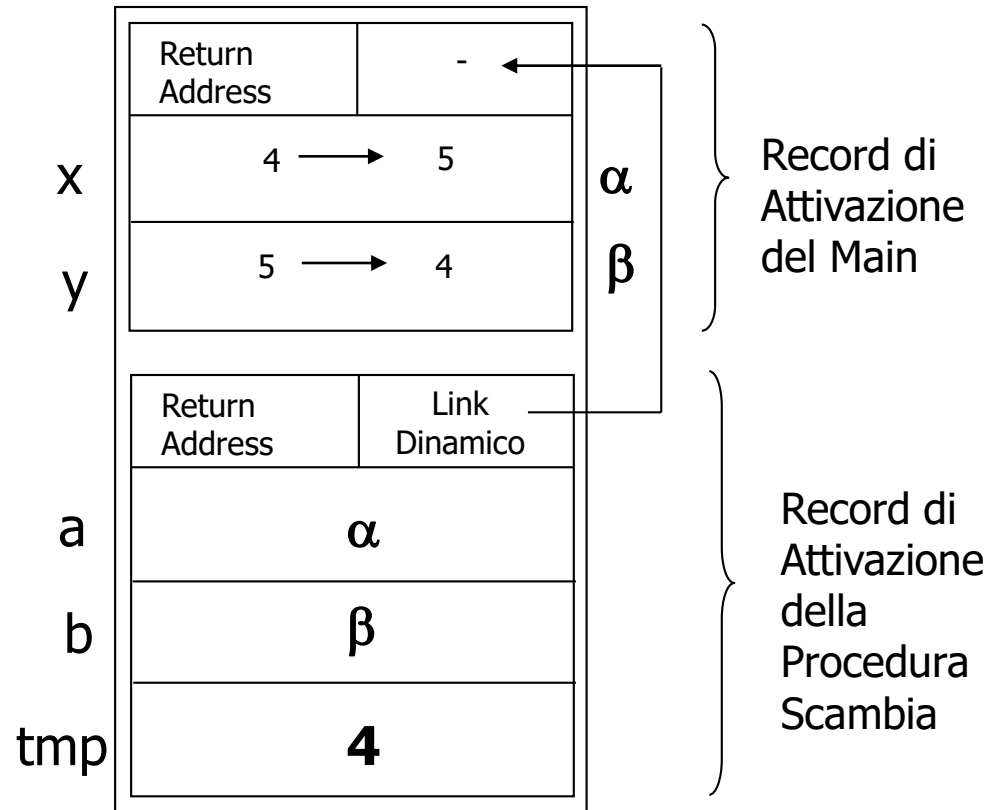
```
#include <stdio.h>

void scambia ( int *, int *);
```

```
int main(void){
    int x=4,y=5;

    scambia(&x,&y);
    printf("%d %d \n",x,y);
}
```

```
void scambia ( int *a, int *b) {
    int tmp=*a;
    *a=*b;
    *b=tmp;
}
```





Passaggio Struct per &

```
#include<stdio.h>
#define C 20
#define FFLUSH while (getchar()!='\n')

void LeggiStringa(char s[], unsigned long dim) {
    unsigned long i;

    for (i = 0; i<dim - 1; i++)
        if ((s[i] = getchar()) == '\n') break;
    if (i == dim - 1) FFLUSH;
    s[i] = '\0';
}

typedef struct {
    char cognome[C], nome[C];
    unsigned short eta;
} persona;
```

```
persona elem;

void LeggiPersona(persona *p) {
    printf("\nInserisci il Cognome ");
    LeggiStringa(p->cognome, C);
    printf("\nInserisci il Nome ");
    LeggiStringa(p->nome, C);
    printf("\nInserisci l'eta' ");
    scanf("%hu", &p->eta);
}

void StampaPersona(persona x) {
    printf("\nCognome = %s ", x.cognome);
    printf("\nNome = %s ", x.nome);
    printf("\nEta' = %hu ", x.eta);
}

int main(void) {
    LeggiPersona(&elem);
    StampaPersona(elem);
}
```



Passaggio di Parametri Vettore

- i parametri formali passati per valore relativamente ad un vettore, si esprimono nelle tre forme equivalenti:

tipobase * nome_parametro;

tipobase nome_parametro[];

tipobase nome_parametro[dimensione];

- tale regola vale sia nel caso in cui il vettore è definito secondo la sintassi del linguaggio C:

tipobase nome_vettore[dimensione],

sia nel caso in cui il vettore sia stato allocato in modo dinamico nell'heap, tramite il comando:

nome_vettore=(cast)malloc(dimensione vettore);



Passaggio di Parametri Vettore

```
#include<stdio.h>
#include<stdlib.h>
#define FFLUSH while(getchar()!='\n')

float *vett;
unsigned long n;

void riempi(float v[], int dim){
    unsigned long i;
    for (i=0; i<dim; i++) {
        printf("Inserisci l'elemento di indice %u ",i);
        scanf("%f",v+i); //scanf("%f",&v[i]);
        FFLUSH;
    }
}

void stampa(float v[], int dim){
    unsigned long i;
    for (i=0; i<dim; i++)
        printf("\nElemento di indice %u = %f ",i,v[i]);
}
```

```
float max(float v[], int dim){
    unsigned long i;
    float max=v[0];

    for (i=1; i<dim; i++)
        if (v[i]>max) max=v[i];

    return max;
}

int main(void)
{
    printf("\nInserisci la dimensione ");
    scanf("%u",&n);

    vett=(float *)malloc(n*sizeof(float));

    riempi(vett,n);
    stampa(vett,n);
    printf("\nMassimo elemento = %f ",max(vett,n));
}
```

Passaggio di Parametri Vettore

```
#include<stdio.h>
#define N 6

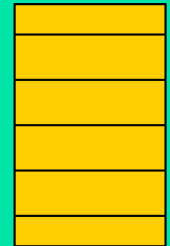
float vett[N]; ← Poteva essere anche dinamico

void riempi(float v[], unsigned long dim){
    unsigned long i;
    for (i=0; i<dim; i++) {
        printf("Inserisci elemento %u ",i);
        scanf("%f", v+i);//scanf("%f",&v[i]);
    }
}

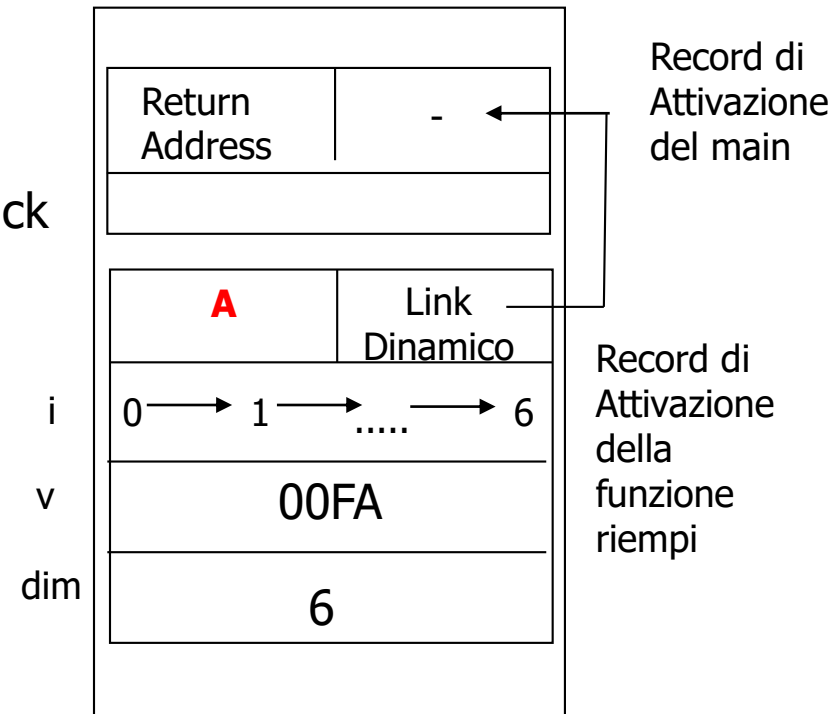
int main(void)
{
    riempi(vett,N);
    A printf(" Ciao ");
}
```

Area
Statica

vett=00FA



Stack



Variabili Locali Statiche

```
#include <stdio.h>
#define N 10
void prova (int);

int main(void)
{
    int i;
    for (i=0;i<N; i++)
        prova(i);
}

void prova (int a){
    static int x=0;
    x+=a;
    printf("\n%d ", x);
}
```

